NOTICE: When government or other drawings, specifications or other data are used for any purpose other than in connection with a definitely related government procurement operation, the U. S. Government thereby incurs no responsibility, nor any obligation whatsoever; and the fact that the Government may have formulated, furnished, or in any way supplied the said drawings, specifications, or other data is not to be regarded by implication or otherwise as in any manner licensing the holder or any other person or corporation, or conveying any rights or permission to manufacture, use or sell any patented invention that may in any way be related thereto.

RADC-TDR-62-511

Final Report

DESIGN AND INSTRUMENTATION OF ERROR-CORRECTING CODES

Prepared for:

ROME AIR DEVELOPMENT CENTER
AIR FORCE SYSTEMS COMMAND
UNITED STATES AIR FORCE
GRIFFISS AIR FORCE BASE, NEW YORK

CONTRACT AF 30(602)-2327

By: *Bernard Elspas*

STANFORD RESEARCH INSTITUTE

MENLO PARK, CALIFORNIA

SRI

STANFORD RESEARCH INSTITUTE

MENLO PARK, CALIFORNIA

October 1962

RADC-TDR-62-511

Final Report

DESIGN AND INSTRUMENTATION OF ERROR-CORRECTING CODES

Prepared for:

ROME AIR DEVELOPMENT CENTER
AIR FORCE SYSTEMS COMMAND
UNITED STATES AIR FORCE
GRIFFISS AIR FORCE BASE, NEW YORK

CONTRACT AF 30(602)-2327

By: *Bernard Elspas*

SRI Project No. 3318

Approved:

J. Reid Anderson
J. REID ANDERSON, MANAGER COMPUTER TECHNIQUES LABORATORY

J. D. NOE, DIRECTOR ENGINEERING SCIENCES DIVISION

Copy No............ [illegible]

When Government drawings, specifications, or other data are used for any purpose other than in connection with a definitely related Government procurement operation, the United States Government thereby incurs no responsibility nor any obligation whatsoever and the fact that the Government may have formulated, furnished, or in any way supplied the said drawings, specifications or other data is not to be regarded by implication or otherwise as in any manner licensing the holder or any other person or corporation, or conveying any rights or permission to manufacture, use, or sell any patented invention that may in any way be related thereto.

Qualified requestors may obtain copies of this report from the ASTIA Document Service Center, Arlington Hall Station, Arlington 12, Virginia. ASTIA Services for the Department of Defense contractors are available through the "Field of Interest Register" on a "need-to-know" certified by the cognizant military agency of their project or contract.

DESIGN AND INSTRUMENTATION OF
ERROR-CORRECTING CODES

PUBLICATION REVIEW

This report has been reviewed and is approved.

Approved: Walter R Richard
WALTER R. RICHARD
Project Engineer
Transmission Branch

Approved: Albert Feiner
ALBERT FEINER, Chief
Advanced Development Laboratory
Directorate of Communications

FOR THE COMMANDER: Irving J. Gabelman
IRVING J. GABELMAN
Director of Advanced Studies

FOREWORD

Dr. Robert A. Short, Dr. Jeremy J. Stone, Mr. William K. English, and Mr. David C. Condon, all members of the project team, made the following contributions to the work: Dr. Short carried out much of the work on cyclic codes described in Sec. II; Dr. Stone is responsible for the research on multiple-burst correction codes covered in Secs. III-D and III-E; Mr. English carried out the logical and circuit design, and Mr. Condon executed the mechanical design of MAVERIC.

The assistance and advice of Dr. William H. Kautz and Mr. Donald C. Lincicome at Stanford Research Institute are gratefully acknowledged.

Special thanks are due to Dr. Jack K. Wolf of Rome Air Development Center for suggesting the concept of error-location coding and for his close collaboration on many phases of this portion of the work.

ABSTRACT

This report discusses the properties of cyclic codes and codes related to cyclic codes, particularly with respect to the correction of single and multiple bursts of errors. Tables of these codes are included. Trading relations are developed that relate the number of bursts simultaneously correctible with a multiple-burst-correcting code to the length of these bursts. A bound is derived on the minimum number of check digits required for multiple-burst-correcting codes.

Two new classes of codes, both of them derived from cyclic codes, are applied in a novel way to problems of multidimensional error checking and of error location. Error-location codes are basically codes for error detection that also locate to within a sub-block of the received message where corrupted digits fall. These codes appear to be useful in connection with feedback communication systems.

There is also some discussion of reliability questions for terminal equipment, a description of a versatile encoding and decoding device for cyclic codes (MAVERIC), and a discussion of quasi-cyclic, pseudo-cyclic and shortened cyclic codes (in which it is shown that these concepts are essentially equivalent).

CONTENTS

CONTENTS

ILLUSTRATIONS

TABLES

I INTRODUCTION

A. GENERAL

The present report is a Final Report describing work carried out from July 1960 through June 1962 under Contract AF 30(602)-2327 with the Rome Air Development Center of the Air Force Systems Command. An earlier Interim Technical Report (RADC TR 61-259[1]* and Supplement 1[2] thereto) covering the work of the first year, was issued in October 1961.

The results of the first year's efforts will not be repeated in detail in this Final Report, since the details are available in the Interim Report. However, specific reference will be made to the early work, where appropriate, to facilitate the reader's obtaining an over-all picture. We also include in this section an abstract of the Interim Report.

B. BACKGROUND OF THE PROBLEM

The objective of the research was the development of coding methods and instrumentation that will nullify or minimize the effect of circuit noise, or other disturbances, on the transmission of digital information.

The problem of reliable transmission of digital data over various kinds of channels is one of long standing. The theoretical capabilities of communication channels are described by the classical results of Shannon's theory of information, known since 1948. And since then, much effort has been devoted toward making real communication systems perform in a manner that is consistent with Shannon's fundamental theorem for the noisy channel.[3] According to this theorem, there exists for any channel a measure, C, of channel capacity, and that for any rate R less than C, there exists a method of encoding data for the channel such that information may be sent over the channel at a rate of R bits per second with an arbitrarily small probability of error.

*References are listed at the end of the report.

No practical channel has yet been used in a manner consistent with the expectations this theorem aroused. The catch was that, in order to operate at rates very near the channel capacity with extremely small probabilities of error, fantastically complex (and hence expensive) coding schemes appeared to be needed, even for relatively simple, idealized channels. The work of Elias[4,5] and Shannon[6] on error probability bounds indicated this need as long ago as 1955. Hence, even today practical communication systems achieve a low probability of error (reliability) on!y by operating at data rates that are a small fraction of the theoretical channel capacity.

For a real, continuous channel, a digital data source may be coded in two (or possibly more) distinct successive operations: redundant digital coding, followed by modulation of the signal. The first step has also been called *source encoding* and the second *channel encoding*. The emphasis in this research program has been on digital coding.

In the usual binary case, source encoding takes k-binary-digit blocks from the source and maps them into n-digit blocks to be processed by the modulator, where $n > k$. The bounds of Elias and Shannon indicate that the message block length, n, must be of the order of hundreds or even thousands of binary digits in order to obtain error probabilities of, say, 10^{-5} at rates of around 90 percent of channel capacity with even moderately noisy channels. The complexity of digital encoding and decoding equipment is directly related to the size of the data blocks they process.

This relationship between size of data block and complexity of equipment is also true for the channel encoding equipment, which modulates the signal on an analog, rather than a digital, basis. There is a strong feeling that the increasing use of, and inherent ease of mass-producing, digital components will continue to make digital techniques more significant than analog modulation techniques. (See also Chapter 1 of Ref. 7 for a discussion of this point.)

Until very recently, there were no known digital coding schemes for blocks hundreds of binary digits long that were efficient and that could be instrumented economically. This situation has changed with the advent of cyclic codes possessing a high degree of algebraic structure. Among these cyclic codes are the most efficient known multiple-error-correcting codes (those of Bose and Ray-Chaudhuri[8,9]), as well as the burst-error-correcting codes of Abramson,[10] Fire,[11] Melas,[12] Elspas,[13] and others. It is the algebraic structure of these codes that permits economical instrumentation, and that makes possible precise mathematical statements about their capabilities.

This break-through in coding theory means that quite efficient codes with block lengths of hundreds (and in some cases even thousands) of binary digits can now be instrumented with special-purpose encoding and decoding equipment costing on the order of ten thousand dollars or less, instead of with a large-scale high-speed computer, which would be required for a comparably large code lacking algebraic structure.

Other recent developments have been made in coding that use statistical decoding techniques (sequential decoding).[7] These techniques have not been considered in the present study. They are under intensive investigation, principally at the Lincoln Laboratory of MIT. Future developments along such statistical lines may become increasingly important to coding theory.

C. ABSTRACT OF THE INTERIM TECHNICAL REPORT

The interim technical report develops the properties and instrumentation of cyclic codes, with particular reference to their burst-error correction capabilities. Necessary and sufficient conditions for optimum burst-error correcting codes are derived and used to find all such optimum codes of practical size. These optimum codes and a number of other (non-optimum) ones are tabulated.

Consideration is given also to the implementation of the encoding and decoding operations for cyclic codes in terms of logical circuits. Several types of circuits are exhibited for both encoding and decoding. Correction of random errors, correction of bursts of errors, and error detection are all taken into account.

Equations are derived for the performance of burst-error correcting codes in terms of message error probability (after correction) on a binary burst-error channel behaving according to the Gilbert, two-state Markov model. A three-state model is proposed which is a generalization of the Gilbert model, and in which error bursts are themselves grouped into clusters. This three-state model is shown to be amenable to the same mathematical treatment as the two-state model; in particular, the same performance equations may be used.

Finally, two mathematical results are presented which permit the construction of codes correcting several bursts of errors per message block. The first result relates the multiple-burst correction capability of a cyclic binary code of known minimum distance (*e.g.*, the Bose-Chaudhuri codes) to its random-error correction capability. The second result is concerned with multiple-symbol-burst-correcting codes over $GF(p^k)$, where for cases of interest p is a prime greater than 2.

D. SUMMARY

1. Single-Burst Error-Correcting Codes

Section II of this report is concerned with the extension of the earlier results (reported in Sec. II of the Interim Report) to longer and more complex cyclic codes. The earlier work concentrated on finding all optimum burst-error correcting cyclic codes of reasonable size, but these turned out to be limited to rather short bursts (four binary digits or less).

Codes for much longer bursts are tabulated in Table I of this report; these are, of course, nonoptimum codes, but many of them are not far from optimum, and they are believed to be quite useful.

2. Multiple-Burst Error-Correcting Codes

Some initial results on codes for the correction of several independent burstsof errors were reported in Sec. V of the Interim Report. These early results are restated briefly in Sec. III-B of the present report.

Considerably more work was done on this problem resulting in four new approaches, which are also described in detail in Sec. III.

The first of these new approaches depends on the concept of interlacing codes. It leads to multiple-burst correction codes with particularly simple structure, although they are, in general, not very efficient.

The second new approach makes use of the Chinese Remainder Theorem (from mathematical number theory) in an ingenious way. The resulting codes are not cyclic, and their implementation poses some difficulties, but they are, in general, quite efficient.

The third approach makes use of codes over a nonbinary symbol alphabet (q-nary codes, where q is a power of 2) to achieve multiple-burst error correction. These codes are intimately related also to the Chinese Remainder Theorem codes.

The fourth approach depends on empirical test procedures (too tedious for all the simplest codes, except with the aid of a digital computer). Only 19 codes were subjected to these tests, but the results are instructive in themselves, and lead in fact to several useful codes.

All of the above code construction techniques lead to trading relations between the number of bursts and the lengths of the bursts that are correctible with the corresponding codes. The trading relations obtained are similar in general form, although they differ in detail.

3. Multidimensional Burst Error Correction

Section IV of this report treats a problem which was not considered in the early project work. Consequently no discussion of this topic will be found in the Interim Report.

Some channels involve the transmission of digital data in a two-dimensional (or higher-dimensional) format. The serial-parallel transmission of data over multiplexed channels is one example where the format is conceptually two dimensional. In other cases, for example, in multi-track magnetic tape recording, the *physical* format is actually two dimensional.

It is conceivable that some channels of this sort may be subject to noise bursts that exhibit strong correlation in both dimensions, *e.g.*, in time and frequency. For these channels, it is reasonable to inquire whether some sort of two-dimensional burst-error correction is possible. Section IV describes the results of an inquiry along these lines, and gives some definitive results for the "spot" correction capabilities of a class of codes which are cyclic in two dimensions. All of these results generalize readily to higher dimensions.

4. Error-Locating Codes

The bulk of the work on this project has been concerned with codes for the *correction* of errors, as opposed to error detection codes.

Section V, on the other hand, discusses a new concept* in error control, called "error location coding." The basic idea is to augment the capability for error detection to permit the (rough) localization of errors within a received message, at least to within a sub-block of that message. Thus, this kind of coding is intermediate to error detection and error correction. It provides an attractive alternative to error detection in feedback communication systems where detected errors induce a repeat transmission of the mutilated portion of the message. The advantage of error location coding (over straight error detection) is that the net data rate be kept higher—even with long block lengths—since the sub-blocks may be short compared to the over-all block.

Section V describes the structure of several families of error-locating codes. Mathematical proofs of the properties of these codes appear in Appendix I.

5. MAVERIC - A Versatile Encoder-Decoder for Cyclic Codes

As part of the work on this contract, there was designed and constructed a versatile encoder-decoder for cyclic codes. The acronym, MAVERIC, applied to it, stands for Magnetic Versatile Information Corrector. The design of the machine is based largely on the encoding and decoding instrumentation concepts described in Sec. III of the Interim Report.

The reduction of these ideas to a piece of laboratory hardware was accomplished by making heavy use of magnetic circuit techniques, a field in which this Laboratory has particular competence, and which is particularly appropriate to this kind of application. The circuit functions required (principally shift registers and half-adders) are conveniently realized with multiaperture core techniques, and the speed capabilities of magnetic cores are well within the requirements of such a device.

Section VI of this report describes in general terms the structure and operation of MAVERIC, without going into extreme details of logic or circuitry.

The machine itself was of considerable use in the testing of the cyclic codes described in Sec. II of this report, and also in the formulation and study of reliability questions concerning encoding and decoding equipment.

*Suggested by J. K. Wolf (private communication).

E. PUBLICATIONS RESULTING FROM THE PROJECT

The following journal publications written by project personnel have resulted from this research effort:

1. J. J. Stone, "Multiple-Burst-Error Correction," *Inf. and Control* 4, pp. 324-331, December 1961.
2. B. Elspas and R. A. Short, "A Note on Optimum Burst-Error-Correcting Codes," *IRE Trans. PGIT*-8, pp. 39-42, January 1962.
3. J. J. Stone, "Multiple-Burst-Error Correction with the Chinese Remainder Theorem," *J. Soc. for Indust. Appl. Math.* (in press).

The following paper, co-authored by the project leader, has been submitted for publication:

B. Elspas and J. K. Wolf, "Error Location Codes—A New Concept in Error Control," submitted to *IRE Trans. PGIT*, Correspondence section.

In addition to the above publications, the following lectures were given by project personnel on research results of this work:

B. Elspas and W. K. English, "Error-Correcting Codes and One Technique for their Implementation," before the San Francisco Chapter of the IRE Professional Group on Electronic Computers, May 1962.

B. Elspas, lectures on "Optimum Burst-Error-Correcting Codes" "Multidimensional Error Correction," and on "Multiple-Burst-Error Correction," as part of the University of Michigan Summer Engineering Conferences, June 1962.

II SINGLE-BURST ERROR-CORRECTING CODES

A. INTRODUCTION

Codes for the correction of a single burst of errors occurring within a code block were extensively discussed in the Interim Technical Report under this contract.[1] The groundwork for the study of the burst-error correction properties of cyclic codes was laid in that report, particularly in Secs. I and II. Consequently, a knowledge of the pertinent background material will be assumed in this sequel.

Cyclic codes are a subclass of group codes with particularly useful properties both in regard to their mathematical structure and in regard to their ease of implementation. The pertinent parameters of any group code are:

(1) The block length, n;

(2) The number of information places per block, k; and

(3) The number of redundant (check) places, $r = n - k$.

The discussion in this section will be limited to binary codes.

The burst-error correction capability of a code can be characterized by an integer, b, defined as the length (span) of the longest burst of errors such that all error patterns of length not exceeding b can be corrected by the code. In general, burst-error correcting codes will also be capable of correcting some (but not all) error patterns of length greater than b. Usually, however, such capability is of rather limited utility; therefore, one is justified in summarizing the burst-correction capability of a code in terms of the single parameter, b.

In the earlier report cited above, a number of inequalities relating b to the other code parameters were derived. Perhaps the most important of these is the upper bound on b:

$$b \leq [r/2]$$

Another important bounding relation is the upper bound on *n*:

$$n \leq 2^{r-b+1} - 1$$

When this bound is met exactly, one speaks of an *optimum* burst-error correcting code. In Sec. II of the Interim Report, a tabulation was given for all optimum burst-error correcting codes of length less than $n = 4096$, using up to 16 check digits. These codes are for bursts of length $b = 2$, 3, and 4. It was also shown that within the above range, no optimum $b = 5$ codes exist. It is now known also that there are no optimum $b = 6$ codes within that range.

While the number of optimum burst-error correcting codes is rather limited, there do exist a large number of codes that are nearly optimum, and whose parameters *n, k,* and *r* fall within the practical range. The main purpose of this section is to present results on these codes. These results are presented in Table I, listing the values of the parameters, *n, k, r* and *b* of these codes against the defining polynomial, $g(x)$. For many of the tabulated codes, the minimum Hamming distance, *d*, between code words is also known, and in those cases this parameter is also tabulated. A code with (odd) distance $d = 2t + 1$ is capable of correcting *t* random errors per block, while a code with (even) distance $d = 2t + 2$ will correct any *t* random errors, and also detect the occurrence of $t + 1$ errors per block simultaneously.

The results shown in Table I were obtained in some cases by analytical techniques (see the Interim Report), but are largely the product of empirical testing techniques carried out with the aid of B.U.R.P.(see Appendix C of the Interim Report), or the error-correcting encoder-decoder, MAVERIC, described in Sec. VI of the present report.

B. USE OF THE TABLE OF SINGLE-BURST ERROR-CORRECTING CODES

The codes in Table I are listed in increasing order of block length *n* from $n = 3$ to $n = 635$, and within each section (of constant *n*) in increasing order of redundancy, *r*. Thus the codes with highest information rate, k/n, appear at the beginning of each section. Since all of the codes listed are cyclic, they are specified by giving the generating polynomial, $g(x)$. The convention used in listing these polynomials is to express the (binary) coefficients of the polynomials (listed in descending powers of *x*) as an octal number. Thus, the octal code, "473,"

Table I

SINGLE-BURST ERROR-CORRECTING CODES

n	k	r	b	POLYNOMIAL*	d	COMMENTS
3	1	2	1	7	3	Cyclic Hamming
5	1	4	2	37	5	
7	4	3	1	13	3	Cyclic Hamming
7	3	4	2	27	4	Abramson
7	1	6	3	177	7	
9	3	6	3	111	3	Interlace
9	1	8	4	777	9	
15	10	5	2	65	4	Abramson
15	9	6	3	171	4	Melas
15	7	8	4	721	5	Bose-Chaudhuri
15	6	9	4	1163	6	
15	5	10	5	2467	7	Green-San Soucie (also corrects all double errors + b = 5)
15	5	10	5	2041	3	Interlace
15	4	11	5	7531	8	
15	3	12	6	11111	5	Interlace
17	9	8	3	471	5	
17	8	9	3	1513	6	
21	16	5	1	61	3	
21	15	6	2	123	4	
21	15	6	2	127	3	
21	14	7	3	171	4	
21	13	8	3	645	3	
21	13	8	2	575	4	
21	12	9	4	1663	5	Also corrects double errors + b = 3
21	12	9	4	1357	4	
21	12	9	2	1607	4	
21	12	9	4	1101	3	
21	11	10	4	3303	4	
21	11	10	4	2325	6	
21	10	11	5	7707	4	
21	10	11	4	5031	5	
21	9	12	5	15533	3	
21	9	12	6	10111	4	Interlace
21	9	12	5	17053	8	
21	9	12	6	14515	6	
21	8	13	6	25727	6	Fire
21	8	13	5	26755	6	
21	7	14	7	47343	8	

* Octal form: thus "473" = 100111011 stands for $x^8 + x^5 + x^4 + x^3 + x + 1$

Table 1 *Continued*

n	k	r	b	POLYNOMIAL	d	COMMENTS
21	7	14	7	40201	3	Interlace
21	6	15	7	140603	6	
21	6	15	7	173465	7	
21	6	15	7	151445	8	
21	5	16	8	214537	10	
21	4	17	7	542613	9	
21	3	18	9	1647235	12	
21	3	18	9	1111111	7	Interlace (m = 3, b = 3)
21	2	19	9	3333333	14	
21	1	20	10	7777777	21	
23	12	11	5	5343	7	Golay-Prange (perfect)
23	11	12	5	17445	8	
27	9	18	9	1001001	3	Interlace
31	25	6	2	157	4	Abramson
31	21	10	4	3551	5	Bose-Chaudhuri
31	20	11	5	4673	6	Kasami (Bose-Chaudhuri with all-check)
33	23	10	3	2251	$\geq$ 3	*I*
33	23	10	3	3043	$\geq$ 3	*I*
35	27	8	3	553	4	Fire
35	23	12	5	13627	$\geq$ 3	*I*
39	27	12	5	13617	$\geq$ 3	*I*
41	21	20	9	6647133	9	*I* } Mattson
41	21	20	8	5747175	9	*I* } (*d*)
42	20	22	10	25250025	4	Interlace
42	18	24	12	100010101	4	Interlace
42	14	28	14	2025052005	8	Interlace (m = 3, b = 2)
42	10	32	16	40120210525	10	Interlace (m = 4, b = 2)
42	6	36	18	($x \rightarrow x^6$ in 177)	7	Interlace (m = 6, b = 3)
43	29	14	5	52225	$\geq$ 3	*I*
43	29	14	5	64213	$\geq$ 3	*I*
43	29	14	3	47771	$\geq$ 3	*I*
43	28	15	5	134635	$\geq$ 4	
45	33	12	3	10011	3	Interlace
45	27	18	9	1001111	4	Interlace
46	24	22	10	11052005	7	Interlace (m = 3, b = 2)
47	24	23	$\geq$ 7	43073357	11	Mattson (*d*); Stone (*b*)

Table 1 *Continued*

n	k	r	b	POLYNOMIAL	d	COMMENTS
51	43	8	3	433	≥ 3	*I*
51	42	9	3	1455	≥ 4	
51	41	10	4	3501		
51	40	11	4	4703		
51	40	11	4	6547		
51	35	16	7	304251		
55	35	20	9	7164555	≥ 3	*I*
57	39	18	7	1735357	≥ 3	*I*
57	39	18	8	1341035	≥ 3	*I*
57	38	19	8	3443047	≥ 4	
63	56	7	2	305	4	Abramson
63	55	8	3	711	4	Melas
63	54	9	3	1133	4	
63	54	9	3	1537	4	
63	54	9	3	1621	≥ 3	
63	53	10	4	2263	4	
63	52	11	4	6023	≥ 3	
63	51	12	5	16447	4	
63	51	12	4	12471	5	Bose-Chaudhuri
63	50	13	6	22377	≥ 4	
63	50	13	5	37513	6	Bose-Chaudhuri with all-check
63	49	14	6	61303	≥ 3	
63	48	15	7	105437	≥ 5	
63	47	16	7	220425	≥ 6	
63	46	17	8	730535	≥ 5	
63	45	18	8	1371261	≥ 5	
63	44	19	9	2002353	≥ 8	Others known
63	43	20	9	6145045		Others known
65	53	12	3	12345	5	Bose-Chaudhuri
69	36	33	15	101011100011	7	Interlace ($m = b = 3$)
73	64	9	3	1027	≥ 3	*I*
73	63	10	4	2343	≥ 4	
75	55	20	5	4000041	3	Interlace
85	77	8	2	613	3	*I*
85	76	9	3	1501	4	
85	73	12	5	10131		
89	78	11	4	4303	≥ 3	*I*
91	79	12	4	10571	≥ 3	*I*
92	48	44	20	($x \to x^4$ in Golay code)	7	Interlace ($m = 3$, $b = 4$)

Table I *Concluded*

n	k	r	b	POLYNOMIAL	d	COMMENTS
93	83	10	3	2065	≥ 3	*I*
93	82	11	4	6137	≥ 4	
105	94	11	4	5267	4	Fire
105	93	12	5	10555	≥ 3	*I*
105	91	14	6	70521	≥ 4	Kasami
117	105	12	4	13413	≥ 3	*I*
129	115	14	3	42721	≥ 3	*I*
133	115	18	7	1254355	≥ 3	*I*
133	115	18	7	1532007	≥ 3	*I*
133	115	18	6	1302357	≥ 3	*I*
151	136	15	6	114371	≥ 3	*I*
155	145	10	3	2205	4	Fire
195	182	13	5	22475	≥ 4	Kasami
195	183	12	4	15347	≥ 3	*I*
217	205	12	4	11245	4	Fire
217	202	15	6	120247	≥ 3	*I*
255	246	9	2	1455	4	Abramson
255	245	10	3	3523	≥ 3	
255	244	11	3	4765	≥ 4	
255	243	12	4	17667	≥ 3	
255	242	13	4	26531	≥ 4	
255	241	14	5	76305	≥ 3	
255	240	15	5	112471	≥ 4	
255	239	16	6	301565	≥ 3	
257	241	16	3	214461	5	*I*, others known
273	261	12	4	10743	≥ 3	*I*
279	265	14	5	45045	4	Fire
315	304	11	3	4043	4	Fire
465	454	11	3	7275	4	Kasami
511	499	12	4	10451	4	Optimum b = 4
595	581	14	5	64655	4	Kasami (maximum n for given r, b in cyclic code.)
635	623	12	3	10343	4	Fire

stands for the coefficients, 100111011 (grouped in threes), and thus this octal number represents the polynomial, $x^8 + x^5 + x^4 + x^3 + x + 1$. The standard octal representation:

0	=	000	4	=	100
1	=	001	5	=	101
2	=	010	6	=	110
3	=	011	7	=	111

has been employed, with leading zeros suppressed.

In some cases, various "Comments" have been supplied in the extreme right-hand column of the table. These take the form either of naming the originator of the code (*e.g.*, Abramson, Melas, Green - San Soucie, etc.) or of designating to which general class of cyclic codes it belongs, (*e.g.*, irreducible polynomial = I, Bose-Chaudhuri, interlace code, etc.), or of providing other pertinent information. In those cases where the codes are not attributed to a definite researcher (and also in many of the others), the burst-correction parameter, b, was determined by the author and his co-workers. The Bose-Chaudhuri codes, for example, were studied by their inventors for their *distance* (random error correction) properties rather than for burst-error correction. It should be noted, too, that all the codes derived from irreducible polynomials are members of the class of Bose-Chaudhuri codes, although this has not been explicitly indicated in the table.

The codes of Abramson ($b = 2$) and Melas ($b = 3$) are optimum burst-error correcting codes, as is the (511, 499) code shown for $b = 4$. Longer optimum $b = 4$ codes given in the Interim Report (Table II, p. 30), are not repeated here.

The codes marked "interlace" have generating polynomials of the form, $g(x) = f(x^a)$ for some integer a. By virtue of the interlace theorem (see Sec. III-C), such codes will correct bursts of length $b' = ab$, over a block length $n' = an$, if $f(x)$ generates a burst-b code of length n.

III MULTIPLE-BURST ERROR-CORRECTING CODES

A. INTRODUCTION

Burst-error correcting codes are useful in connection with communication channels where noise (and hence digit errors) tend to occur in bursts. If the mean period between noise bursts is considerably longer than the code block length, and most of the bursts are shorter than the burst-correction capability, b, of the code, then one may expect that the code will correct the most probable error patterns. On the other hand, in order to find a reasonably efficient code, one may be forced to use a block length comparable to (or even in excess of) the mean period between noise bursts. In such cases, several "bursts" of errors may occur within the same code block, and the code may be incapable of correcting these errors. This will occur whenever the over-all span of the errors exceeds the capability, b, of the code.

The above state of affairs suggests that one might investigate whether codes can be designed to correct several independent bursts of errors occurring within the same code block. Such codes, called *multiple-burst error-correcting codes*, are discussed in this section. They may be characterized by two parameters: m, the multiplicity of bursts to be corrected, and b, the length of the individual bursts. One thus speaks of an (m,b) multiple-burst correcting code.

Several approaches to this kind of code have been found, among them the approach of Stone, the use of interlacing, Chinese Remainder Theorem codes, and Reed-Solomon codes.

B. CODES BASED ON STONE'S THEOREM

The following theorem (due to J. J. Stone) provides one approach to the study of multiple-burst codes. The original statement and proof of this theorem appear in Sec. V-B of the Interim Report. The theorem is restated here (without proof) for the sake of completeness.

> *Theorem 1*—If C is a cyclic binary code of length n and minimum distance $d = 2mt + 1$, where $n > 3mt$, then C corrects all error patterns made up of up to m bursts of width up to b,

where

$$b = t + [(t - 2)/2 + 3/4m]$$

For $m = 1$, this gives

$$b = t + [(2t - 1)/4] = [(6t - 1)/4]$$

while for $m > 1$, one has

$$b = t + [(t - 2)/2] = [(3t - 1)/2]$$

The above theorem provides a means whereby the random error correction capability of a cyclic code (such as the Bose-Chaudhuri codes) can be "converted" into a capability for correction of multiple bursts of errors. In particular, the case $m = 1$ corresponds to single burst error correction, while the opposite extreme, $b = 1$, corresponds to the correction of random isolated errors (bursts of width one).

A central point, which must not be overlooked here, is that the same code can be used for the correction of more or fewer bursts depending on their length. For example, any cyclic code of weight 25 (a 12-error correcting code) and length $n > 36$ can be exploited to correct up to m bursts of length b (or less) for the parameter combinations shown below.

m	b	t
1	17	12
2	8	6
3	5	4
4	3	3
6	2	2
12	1	1

The relation of m *vs.* b stated in Theorem 1 thus provides a trade-off relation between burst length and number of bursts to be corrected. We shall see that other schemes for the construction of multiple-burst codes also possess similar trading relations.

It should be noted that the assertion contained in Stone's theorem is of the nature of a (lower) bound on performance of a given code. In specific cases, one finds that the performance (in terms of m and b) actually exceeds the stated capability. However, the stated capability is all that one can guarantee.

C. INTERLACED CODES

1. THE INTERLACE THEOREM

The concept of interlacing codes for the provision of multiple-burst correction capability is dependent on the following interesting theorem (due in part to F. Corr[14]).

> *Theorem 2 (Interlace theorem)*—Let C_1be a cyclic (n, k) code generated by the polynomial $f(x)$ of degree $r = n - k$ over $GF(2)$. If C_1 is capable of correcting t random errors, or, alternatively is capable of correcting single bursts of length up to b, then for any integer a, the polynomial $g(x) = f(x^a)$ generates a cyclic code C_a of length na with ka information places, and the code C_a can correct:
>
> (a) Up to t bursts of errors, each of width up to a; or
>
> (b) Any single burst of length up to ba.

Proof: Since $f(x)$ generates a cyclic code of length n, we have that $f(x)$ divides $x^n + 1$. Consequently, $g(x) = f(x^a)$ divides $(x^a)^n + 1 = x^{na} + 1$, so that C_a is a code of length na. Since the degree of $g(x)$ is $a(n - k) = ar$, the code C_a has ra check digits per block.

The code C_a may be interpreted in the following interesting fashion: Imagine an array of a independently chosen n-digit words from the original code, C_1. Now suppose that one transmits these na digits serially, by first transmitting the first digits of each of the n-digit words (in turn), then the second digits of each word, and so forth. We have in effect *interlaced* these a words. It is now asserted that the set of all na-digit messages obtained in this way is precisely the code C_a. For if $u^{(i)}(x)$; $i = 1, \ldots, a$ are each code words of C_1, so that $f(x)$ divides each $u^{(i)}(x)$, then it is easily seen that the na-digit interlaced word is given by the polynomial, $u(x) = u^{(1)}(x^a) + xu^{(2)}(x^a) + \ldots + x^{(a-1)}u(x^a)$. But this polynomial is evidently divisible by $f(x^a) = g(x)$, and hence defines a code word in C_a.

In order to prove assertion (a), examine the interlace structure shown below for the case $a = 3$, $n = 7$. The same proof holds for arbitrary a and n.

0	3	6	9	12	15	18
1	4	7	10	13	16	19
2	5	8	11	14	17	20

It is clear that any t or fewer bursts each of length a (=3) or less can alter at most t digits in any row of this array. But by assumption, the code C_1 (from which the rows are chosen) can correct up to t independent errors per block of n. Hence the code C_a will correct t bursts of width a each.

Assertion (b) follows in similar fashion, since any burst of length ab or less will alter at most b consecutive digits in any one row. Since the code C_1 corrects bursts of width b, the assertion follows. This completes the proof of Theorem 2.

The following corollary to Theorem 2 is also pertinent to the construction of multiple-burst codes.

> *Corollary:* If the code C_1 in Theorem 2 is capable of correcting up to t bursts of errors each of width b or less, then the interlaced code C_a will correct up to t bursts of errors, each of width ab or less.

The proof of this corollary is exactly parallel to the proofs of assertions (a) and (b) in Theorem 2.

The above corollary thus gives us a technique for constructing multiple-burst codes for longer bursts in terms of multiple-burst codes for short bursts.

2. Examples Of Interlace Codes

We give here several examples of interlace codes derived from short, efficient cyclic codes of known distance and known burst-correction capability.

Example (a)—Starting from the cyclic (17, 9) code with $g(x) = 1 + x^3 + x^4 + x^5 + x^8$, which has $d = 5$ and $b = 3$, one obtains a family of ($17a$, $9a$) codes tabulated below for $a = 1, 2, 3$.

a	n	k	r	m	b	r^*
1	17	9	8	1	3	6
				2	1	8
2	34	18	16	1	6	12
				2	2	12
3	51	27	24	1	9	18
				2	3	15

The column headed "r^*" shows the theoretical minimum number of check digits required for the indicated correction capability. (See Appendix III).

Example (b)—Starting from the (Golay) code with parameters, $n = 23$, $k = 12$, and generating polynomial, $g(x) = 1 + x + x^5 + x^6 + x^7 + x^9 + x^{11}$, where $d = 7$ and $b = 5$, one obtains the family:

a	n	k	r	m	b	r^*
1	23	12	11	1	5	10
				3	1	11
2	46	24	22	1	10	20
				3	2	17
3	69	36	33	1	15	30
				3	3	22
4	92	48	44	1	20	40
				3	4	26

Other examples of interlaced multiple-burst correction codes are listed in Table I.

D. CHINESE REMAINDER THEOREM CODES*

1. Summary

The Chinese Remainder Theorem of number theory describes conditions under which a number may be recaptured by the knowledge of the residues to which it gives rise when divided by certain moduli. Thus, under conditions that allow recapture, a number might be communicated from sender to receiver by the transmission of its residues. If additional residues were sent, the number might be communicated, despite some disruption of the transmission. By using these ideas and a generalization of the Chinese Remainder Theorem, which deals with polynomials over the Galois fields $GF(p^n)$ instead of numbers, a method is given for transmitting information which seems suitable for multiple-burst-error correction. The Reed-Solomon Polynomial Codes are shown to be a special case of these codes.

* The material in this section was developed and prepared by J. J. Stone.

2. Definitions And Background

If a, b, and m are integers, define a to be congruent to b modulo m, denoted $a \equiv b \pmod{m}$, if and only if $m|(b-a)$. Two integers m_1 and m_2 are said to be relatively prime if $d|m_1$ and $d|m_2$ implies $d = \pm 1$. The relevant part of the Chinese Remainder Theorem in number theory follows:

> *Theorem 3 (Chinese Remainder Theorem)*—Let m_1, m_2, ..., m_r be integers that are relatively prime in pairs. Let
>
> $$M = \prod_{i=1}^{r} m_i$$
>
> denote their product. If a_1, a_2, ..., a_r are any given integers, then there exists one and only one number f such that $0 \leq f < M$ and
>
> $$f \equiv a_i \pmod{m_i} \; ; \quad i = 1, 2, \ldots, r \quad . \tag{1}$$

Proof: (See Uspensky and Heaslet.)[15] In particular, if one knows that for some moduli m_{i_0} an unknown number f satisfies

$$0 \leq f < \prod_{i=1}^{r} m_i$$

and if one knows the residues a_i, $0 \leq a_i < m_i$, which result when f is divided by m_i, then by Theorem 3, f can be recaptured. In fact a formula for f may be given. Let t_{i_0} be an integer satisfying

$$\frac{M}{m_{i_0}} t_{i_0} \equiv 1 \pmod{m_{i_0}} \tag{2}$$

(The existence of such integers is assured by the assumed relative primeness property of the m_i.) Then f is the only common solution of the following two conditions:

$$f \equiv \frac{M}{m_1} t_1 a_1 + \frac{M}{m_2} t_2 a_2 + \ldots + \frac{M}{m_r} t_r a_r \tag{3(a)}$$

$$0 \leq f < M \quad . \tag{3(b)}$$

In other words if the m_i, t_i and M are thought of as given constants, to find f from a given set of residues a_i, one need only substitute the a_i in the right hand side of Eq. [3(a)] and divide the result by M. The residue after this division satisfies Eqs. [3(a) and 3(b)] and is f.

We proceed to state a generalization of this theorem that is well known. Let $GF(q)$ be the Galois Field with $q = p^n$ elements. Let $GF(q)[x]$ be the ring of polynomials over $GF(q)$. Two polynomials $m_1(x)$ and $m_2(x)$ will be called relatively prime if $g(x) | m_i(x)$ $i = 1, 2$, and $g(x)$ in $GF(q)[x]$ implies $g(x)$ is a constant polynomial. If $a(x)$, $b(x)$ and $m(x)$ are in $GF(q)[x]$ then we define $a(x)$ is congruent to $b(x)$ modulo $m(x)$, denoted $a(x) \equiv b(x) \bmod m(x)$ if and only if $m(x) | [b(x) - a(x)]$. The following theorem can now be stated.

> *Theorem 4*—Let $m_1(x)$, $m_2(x)$, ..., $m_r(x)$ be in $GF(q)[x]$ and relatively prime in pairs. Let
>
> $$M(x) = \prod_{i=1}^{r} m_i(x)$$
>
> denote their product. If $a_1(x)$, $a_2(x)$, ..., $a_r(x)$ are any given members of $GF(q)[x]$, then there exists one and only one member $f(x)$ of $GF(q)[x]$ such that the degree of $f(x)$ is less than the degree of $M(x)$ and
>
> $$f(x) \equiv a_i(x) \bmod m_i(x) \qquad (4)$$

Proof: The proof uses the fact that $GF(2)[x]$ is a Gaussian domain, (see Jacobson[16]) and it is analogous to the proof of Theorem 1.

The discussion that follows Theorem 3 on the recapturing of f can be applied to this case if f, M, m_i, t_i, a_i are all thought of as polynomials over $GF(q)$ and if the inequality between f and M is reinterpreted as an inequality between their degrees.

If x is a number, $[x]$ denotes the greatest integer less than or equal to x. $V_n[GF(q)]$ denotes the n-dimensional vector space over $GF(q)$.

3. Numerical Encoding Over $GF(q)$

Associate with each element of $GF(q)$ (in a one-to-one fashion) an integer $I(u)$, $0 \leq I(u) < q$. If a member of $V_n[GF(q)]$, $u = (u_0, u_1, u_2, \ldots, u_{k-1})$ is to be sent, associate with this vector the number f defined by

$$f = \sum_{i=0}^{k-1} I(u_i) q^i$$

Let $m_1, m_2, \ldots, m_r$ be a set of relatively prime integers such that

$$M = \prod_{i=1}^{r} m_i > q^k \quad .$$

Let a_i be the residue of f when divided by m . Denote $[\log_2 m_i]$ by $d_i - 1$. This represents the highest power of two necessary to represent m_i in binary form. Let a_i be represented in binary notation,

$$a_i = \sum_{j=0}^{d_i - 1} b_{ij} 2^j$$

In place of the original block u, we send the binary representation in order as follows:

$$v = (b_{10}, b_{11}, b_{12}, \ldots, b_{1,d_1-1}, b_{20}, b_{21}, b_{22}, \ldots, \\ b_{2,d_2-1}, \ldots, b_{r0}, b_{r1}, \ldots, b_{r,d_r-1}) \qquad (5)$$

The decoding of v, assuming the block has been transmitted without error, is done by a receiver who knows the m_i being used, the length n of the blocks being sent, and the function I. Theorem 3 allows the receiver to recapture f and after expanding f in the number system with base q; knowledge of I allows him to recapture u.

4. Polynomial Encoding And Decoding Over $GF(2)$

To simplify the discussion, this section assumes that the symbols to be sent are drawn initially from $GF(2)$. If this were not so, a suitable encoding of the symbols could be made. Alternatively, if n method of transmitting q symbols were available, $GF(q)$ might be substituted for $GF(2)$ everywhere to get the appropriate analogy.

Let $u = (u_0, u_1, \ldots, u_{k-1})$ be in $V_k[GF(2)]$. We denote by $f_u(x)$ the polynomial

$$\sum_{i=0}^{k-1} u_i x^i \quad .$$

Let $m_i(x)$, $i = 1, 2, \ldots, r$ be in $GF(2)[x]$, relatively prime in pairs and such that the sum of their degrees is greater than $k - 1$. Denote the degrees of $m_i(x)$ by $d_i - 1$. Let $a_i(x)$ be the polynomial residue of $f_u(x)$ when divided by $m_i(x)$. Then

$$a_i(x) = \sum_{j=0}^{d_i - 1} b_{ij} x^j$$

for some b_{ij} in $GF(2)$. In place of the original block u, we send the $a_i(x)$ in order, by sending their coefficients as follows:

$$v = (b_{10}, b_{11}, \ldots, b_{1,d_1-1}, b_{20}, b_{21}, \ldots, b_{2,d_2-1}, \ldots, \\ b_{r0}, b_{r1}, \ldots, b_{rd_r-1}) \quad . \tag{6}$$

The decoding of v, if no error in transmission has occurred, uses Theorem 4 to recapture f_u, assuming knowledge of k and the $m_i(x)$, and u is formed simply from the coefficients of f.

5. Error Correction And Detection By Voting

The procedure outlined here is very natural and is similar to that used for decoding of Reed-Solomon polynomial codes.[17] We restrict ourselves for simplicity to discussing only the polynomial case. The numerical situation is similar.

Assume now that errors in transmission are expected. Some simplifying assumptions regarding the code parameters are shown in Table II. These are introduced purely to simplify the exposition.

Consider a code with the simplified parameters. By Theorem 4, if the residues $a_i(x)$ of any h moduli $m_i(x)$ are selected, then they determine a unique polynomial of degree less than $dh = k$. If these residues have

Table II

NOTATION FOR CHINESE REMAINDER THEOREM CODES

QUANTITY	NOTATION	SIMPLIFIED SITUATION
degree of moduli $m_i(x)$	$d_i - 1$	$d_i = d$
number of residues $a_i(x)$	r	r
length of given block (u)	k	k assume $d \mid k$ and define $h = k/d$
maximum number of residues presumed changed in transmission	c	c
length of block transmitted (v)	$\sum_{i=1}^{r} d_i$	dr

been correctly transmitted, then this unique polynomial must be the correct one. In any case we refer to it as a test polynomial g. There are evidently $\binom{r}{h}$ different ways of determining a test polynomial.

If, at most, c residues $a_i(x)$ have been changed by errors, $\binom{r-c}{h}$ test functions will agree and turn out to be the correct one, f_u. Thus the transmitted block may be thought of as voting at least $\binom{r-c}{h}$ times for the correct test function. We wish to compute the maximum number of votes that could be cast for an incorrect test function. It is possible that $c - 1$ incorrect residues have been changed so as to be the residues (for the appropriate moduli) of the *incorrect* test function determined by $h - 1$ correct residues and one incorrect residue. In this case all the ways of choosing h residues from this conspiracy of $h + c - 1$ residues, that is, $\binom{h+c-1}{h}$ ways, will give the same incorrect answer. If, however, test functions that involve between them h or more correct residues agree, then it is clear that they must be the correct answer. Hence, all sets of agreeing test functions come from no more than $h + c - 1$ residues and the maximum number of votes cast for the wrong function must be exactly $\binom{h+c-1}{h}$. Fixing this election amounts to ensuring

$$\binom{r-c}{h} > \binom{h+c-1}{h} \tag{7}$$

or

$$r - c > h + c - 1 \quad . \tag{8}$$

Substituting for h and rearranging gives

$$rd - k > d(2c - 1) \tag{9}$$

Notice that $R = rd - k$ is the number of added (redundant) bits. Consider now the number of errors that can happen without ever changing more than c residues. If single errors are considered, the answer is clearly c. If a single burst of width b is considered, the answer is

$$b = (c - 1)d + 1 \quad . \tag{10}$$

For multiple bursts of errors, it is easy to check that $[c/2]$ bursts of width $b = d + 1$ could be handled. Other combinations of b and c are also possible. Substituting Eq. (10) into Eq. (9) shows that for single bursts of width $b = (c - 1)d + 1$,

$$R = rd - k > 2(b - 1) + d \quad . \tag{11}$$

Hence when b is large compared to d, R is not much bigger than $2(b - 1)$, [if b is chosen, as in Eq. (10), to be as large as is possible without ever disturbing $c + 1$ residues]. The above facts are summarized in

> *Theorem* 5—If $m_i(x)$, $i = 1, 2, \ldots, r$ are relatively prime moduli of degree d over $GF(2)$ and if members of $V_k[GF(2)]$ are to be encoded where k/d is an integer, then a Chinese remainder code exists, which will correct $[c/2]$ bursts of width $d + 1$ or one burst of width $(c - 1)d + 1$ if
>
> $$\frac{(r + 1)d - k}{2d} > c$$
>
> and detect either of these sets of errors if
>
> $$\frac{(r + 1)d - k}{d} > c$$
>
> with redundancy $R = rd - k$.

An example follows:

Example 1—Let $k = 7$. Let the $m_i(x)$ be in $GF(2)[x]$ and defined by

$$m_1(x) = x^4 + x + 1$$

$$m_2(x) = x^4 + x^3 + 1$$

$$m_3(x) = x^4 + x^3 + x^2 + x + 1$$

$$m_4(x) = x^4 + x^2 + 1 = (x^2 + x + 1)^2$$

$$m_5(x) = x^4 + x^2 + x + 1 = (x + 1)(x^3 + x^2 + 1)$$

so that $d_i = d = 4$, $r = 5$. For $i = 1, 2, 3$, $m_i(x)$ is irreducible and the last two moduli are shown factored into irreducible factors to aid in establishing the relative primeness of the $m_i(x)$. Note that $R = rd - k = 20 - 7 = 13$; hence using Eq. (9), errors which disrupt at most c residues where $13 > 4(2c - 1)$ can be corrected. Hence $c = 2$ and Eq. (10) gives $b = 5$ for a single burst. $u = (1, 0, 1, 0, 1, 1, 0)$ would be sent by considering the polynomial $f_u = 1 + x^2 + x^4 + x^5$ over $GF(2)$ and constructing the residues:

$$a_1(x) = 0$$

$$a_2(x) = 1 + x + x^2$$

$$a_3(x) = 1 + x + x^3$$

$$a_4(x) = x + x^3$$

$$a_5(x) = x^2 + x^3$$

and sending

$$v = (0, 0, 0, 0, 1, 1, 1, 0, 1, 1, 0, 1, 0, 1, 0, 1, 0, 0, 1, 1)$$

u could be recaptured from *v* if a burst of width 5 or less occurred. The parameters of two somewhat more interesting examples are given below.

Example 2—Let $k = 84$. Let the $m_i(x)$ be the 18 7*th* degree polynomials irreducible over $GF(2)$, so that $r = 18$, $d = 7$. The redundancy is $R = rd - k = 126 - 84 = 42$. $c = 3$ is the largest number of residues that can be safely changed; this allows a single burst of width 15.

Example 3—Let $k = 306$. Let the $m_i(x)$ be 54 of the 9*th* degree polynomials irreducible over $GF(2)$, so that $r = 54$, $d = 9$. The redundancy is $R = rd - k = 486 - 306 = 180$. $c = 10$ is the largest number of residues that can be safely changed. This allows protection against a single burst of width 82 or 5 bursts of width at most 10.

For error detection, it is not necessary that the right polynomial receive a plurality of votes but that a wrong polynomial not receive a unanimous vote. This leads to requiring

$$\binom{r}{h} > \binom{h + c - 1}{h} \tag{12}$$

or in analogy to Eqs. (7), (8), and (9), to requiring

$$R = rd - k > d(c - 1) \tag{13}$$

Thus Example 2 would detect a single burst of width 36. Example 3 would detect a single burst of width 172 or 10 bursts of width at most 10.

6. Carrying Out The Voting

In Part D-2 a formula is discussed for the retrieval of f. This formula (in the polynomial case), depends on certain fixed polynomials $t_i(x)$, which in turn depend on the set of moduli under consideration. Since these sets of moduli vary with each vote, it is important to discuss in more detail the construction of the $t_i(x)$.

Let $t_{ij}(x)$ be the solution of least degree (which exists because the $m_i(x)$ are relatively prime in pairs) of

$$m_j(x)t_{ij}(x) \equiv 1 \quad \bmod m_i(x) \tag{14}$$

Then, keeping $m_i(x)$ fixed and multiplying congruencies Eq. (14) for $j = j_2, j_3, \dots j_h$ gives

$$\left[\prod_{k=2}^{h} m_{j_k}(x)\right] \cdot \left[\prod_{k=2}^{h} t_{ij_k}(x)\right] = \frac{M(x)}{m_i(x)} \left[\prod_{k=2}^{h} t_{ij_k}(x)\right] \equiv 1 \mod m_i(x) \qquad (15)$$

where

$$M(x) = m_i(x) \cdot \prod_{k=2}^{h} m_{j_k}(x) \quad .$$

Hence for the set of moduli $m_i, m_{j_2}, \dots, m_{j_h}$, comparison with Eq. (2) reveals that

$$t_i(x) = \prod_{k=2}^{h} t_{ij_k}(x) \quad . \qquad (16)$$

Thus, if r moduli are used, there are $r(r - 1)$ fixed polynomials t_{ij} to be stored. With these polynomials at hand, each vote requires, as in Part D-2, simply multiplications, additions, and reduction modulo $M(x)$.

In general, the voting procedure might be varied in certain ways. Rather than to go to great lengths to ensure correction with probability 1 in the (somewhat less probable) case that at most c errors occur, one might introduce, for efficiency, some uncertainty in the decoding process, as in sequential decoding.[7] Without going into details, we might "predict the election" in advance without waiting for the plurality as soon as certain votes, using representative moduli, showed some pre-arranged lead. Another possibility, which involves introducing uncertainties in the decoding procedure, would be to "fix" the election in a slightly weaker way than was done in Eq. (12). This approach leads to allowing a small probability that the wrong test function could win, but chooses more favorable parameters than would otherwise be possible.

7. Relationship To Reed-Solomon Polynomial Codes

The Reed-Solomon Polynomial Codes[17] can be looked upon as a special case of these Chinese Remainder codes. The former codes would transmit $u = (u_0, u_1, \dots, u_{k-1})$, in $V_k[GF(2^n)]$, $2^n \geq k$, by letting $P(x) = u_0 + u_1 x + \dots + u_{k-1}x^{k-1}$ and sending (in a binary encoded form)

$$[P(0),\ P(B),\ P(B^2),\ \dots,\ P(B^{2^n-2}),\ P(1)] \qquad (17)$$

where 0, B, B^2, ..., B^{2^n-2}, 1, are the elements of $GF(2^n)$. Decoding is done by recapturing P from its values (after transmission) by an interpolation formula that produces a $k - 1th$-degree polynomial "vote" from any k received values. If $P'(\gamma)$ is the received value of $P(\gamma)$, thus the interpolation approach can be looked upon as the simultaneous solution of sets of k polynomial congruencies drawn from

$$\begin{aligned}
f(x) &\equiv P'(0) && \bmod x \\
f(x) &\equiv P'(B) && \bmod x - B \\
&\ \ \vdots && \\
f(x) &\equiv P'(B^{2^n-2}) && \bmod x - B^{2^n-2} \\
f(x) &\equiv P'(1) && \bmod x - 1
\end{aligned} \tag{18}$$

where $f(x)$ is a polynomial over $GF(2^n)$ of degree less than k, (the degree of the product of any k moduli). Such a polynomial will clearly have the required values. Thus, these codes take as their $m_i(x)$ very simple types of moduli, namely the first degree moduli shown in Eq. (18). The residues $a_i(x)$ have simply become values of P.

We note that an important part of Reed and Solomon's results concern the encoding of elements in $GF(2^n)$ as binary sequences while we have here considered in detail only those applications of the Chinese Remainder Theorem where the information enters as a vector over $GF(2)$.

E. MULTIPLE-BURST ERROR CORRECTION WITH REED-SOLOMON CODES

An approach to multiple-burst error correction that is closely related to Chinese remainder coding is obtained by use of the Reed-Solomon codes, or indeed, any codes of predictable Hamming distance over higher order Galois fields of characteristic two.

The Reed-Solomon codes are most easily described as cyclic codes of a special type over $GF(2^s)$. Thus, each symbol (digit) of the code may be regarded as an s-place vector with binary (zero or one) entries. Let α be an element of order n in $GF(2^s)$. Thus α, α^2, α^3, ..., α^{n-1} are assumed to be distinct, while $\alpha^n = 1$. If one constructs the minimal polynomial over $GF(2^s)$ possessing α, α^2, α^3, ..., α^{d-1} as roots, then this is seen to be the polynomial

$$g(x) = (x - \alpha)(x - \alpha^2)(x - \alpha^3) \ldots (x - \alpha^{d-1})$$

The polynomial $g(x)$ is clearly of period n, and thus it generates a cyclic code of block length n over $GF(2^s)$.* Since the degree of g is $r = d - 1$, this code is an $(n, n - d + 1)$ code. Its Hamming distance is obtained by means of the Bose-Chaudhuri bound (since this code may be viewed as a Bose-Chaudhuri code over $GF(2^s)$ with the result, $d = r + 1$.

If one puts $t = [(d - 1)/2]$, this Reed-Solomon code is capable of correcting any t symbol errors. Stated in terms of blocks of binary digits (s bits per block) the code will correct any pattern of bit errors affecting at most t blocks. As with the Chinese remainder theorem codes, this capability implies that

(1) Any single burst of width $(t - 1)s + 1$ bits; or

(2) Any two bursts each of width $([t/2] - 1)s + 1$ bits, or

(3) Any three bursts each of width $([t/3] - 1)s + 1$ bits; and so forth.

In summary, m independent bursts are correctible if each of them is of width not in excess of $([t/m] - 1)s + 1$ bits, since then each burst affects no more than t/m blocks of s bits. We thus have the trading relation,

$$b = \left(\left[\frac{t}{m}\right] - 1\right)s + 1$$

for (m,b) correction with a Reed-Solomon code [more generally, with any symbol-correcting code over $GF(2^s)$ that is capable of correcting any t symbols].

A slight generalization of this result is obtained if one considers also the correction of bursts of varying lengths at the same time. In fact, let the binary error pattern consist of m_i bursts of width

* The above construction requires that n be a divisor of $q - 1 = 2^s - 1$.

$b_i = a_i s + 1$, for $i = 1, \ldots, l$. Then this error configuration will be correctible if

$$\sum_{i=1}^{l} m_i(a_i + 1) \leq t$$

It is instructive to consider some examples of multiple-burst codes obtained from q-nary Reed-Solomon codes.

Example 1—$q = 2^s = 16$, *i.e.*, $s = 4$; $n = q - 1 = 15$.

The block length in bits is $ns = 60$ and the available range for the Hamming distance is $d = 1, \ldots, 15$. Only odd distances $d = 2t + 1$ are considered for convenience and we tabulate

number of binary check digits $= r' = rs$

number of binary information digits $= k' = ks$

b_m = maximum correctible burst length if m independent bursts are to be corrected.

d	t	r'	k'	b_1	b_2	b_3	b_4	b_5	b_6
3	1	8	52	1	0	0	0	0	0
5	2	16	44	5	1	0	0	0	0
7	3	24	36	9	1	1	0	0	0
9	4	32	28	13	5	1	1	0	0
11	5	40	20	17	5	1	1	1	0
13	6	48	12	21	9	5	1	1	1
15	7	56	4	25	9	5	1	1	1

Observe that these codes do not afford any appreciable degree of multiple-burst correction capability until about 80 percent redundancy is introduced. In fact, only single bursts of any length exceeding one bit are correctible until the information rate, k/n falls below 50 percent. The situation is improved by going to longer codes, as illustrated by the next example.

Example 2—Let $s = 6$, $q = 2^6 = 64$, $n = 21$. Then $n' = ns = 126$, and the available range for d is $d = 1, \ldots, 21$.

d	t	r'	k'	b_1	b_2	b_3	b_4	b_5	b_6
3	1	12	114	1	0	0	0	0	0
5	2	24	102	7	1	0	0	0	0
7	3	36	90	13	1	1	0	0	0
9	4	48	78	19	7	1	1	0	0
11	5	60	66	25	7	1	1	1	0
13	6	72	54	31	13	7	1	1	1
15	7	84	42	37	13	7	1	1	1
17	8	96	30	43	19	7	7	1	1
19	9	108	18	49	19	13	7	1	1
21	10	120	6	55	25	13	7	7	1

Note that with an information rate k/n of about 25 percent, these codes manage to achieve correction of four bursts of length 7, two bursts of length 19, or a single burst of length 43. Even for k/n greater than 50 percent, it is possible to obtain correction for two bursts of length 7, or of a single burst of length 25. Obviously, this is much more respectable than the codes of Example 1.

F. SPECIAL MULTIPLE-BURST CODES OBTAINED EMPIRICALLY

A number of special cyclic codes have been investigated empirically in order to determine their multiple-burst correction properties precisely. The codes in question belong to one or another of the following types:

(1) Simplex codes—that is, codes all of whose nonzero code words have the same weight.

(2) Balanced simplex codes—obtained from simplex codes by adjoining the complements of all words in the simplex code (generalized Green-San Soucie codes).

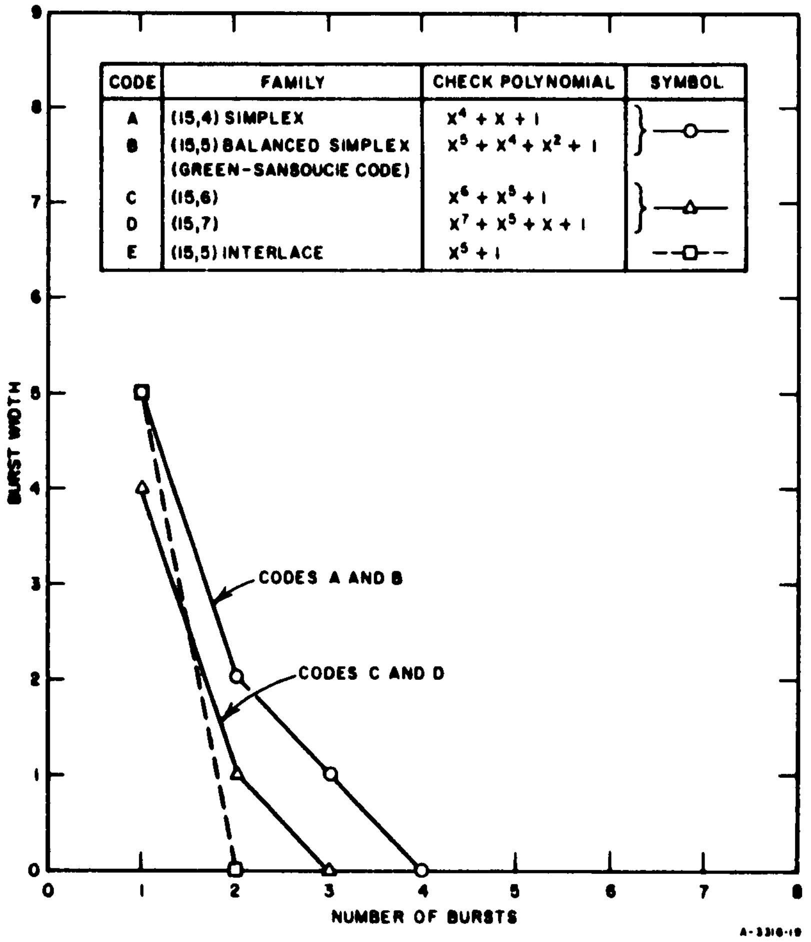
(3) Interlace codes (see Sec. III-C).

The techniques used to determine the results for these codes were principally detailed exhaustion techniques, although in some cases MAVERIC was found to be useful.

The following exhaustion method is particularly appropriate to the simplex and balanced simplex codes. For each value of m, it is desired to determine the largest integer b such that no (nonzero) code word consists of $2m$ (or fewer) bursts of width b (or less). The simplex code words are (except for the all-zero word) all cyclic permutations of a single prototype word (given by a maximum-length shift register sequence). Hence, only one (cyclic) code word need be tested in the case of simplex codes.

The balanced simplex codes are only slightly more troublesome. Here there are two prototype words — a maximum-length shift register sequence and its complementary sequence — that need be tested for any such code. The first few codes in these categories were tested manually, then a simple ALGOL program was developed for this purpose and applied to the remaining simplex and balanced simplex codes.

The *m vs. b* trade-off curves for these codes are shown plotted in Figs. 1 through 6. The codes are identified by means of the *check* polynomial $h(x)$ (instead of the more usual generating polynomial), since $h(x)$ is of lower degree for the codes tested. It is worth noting that the correctible burst width, b_m, falls off more slowly with increasing *m* than one is led to expect from the trade-off relations described in Secs. III-B, D and E. This is probably so because the relations derived earlier are essentially *bounds*, rather than exact performance figures.



CODE	FAMILY	CHECK POLYNOMIAL	SYMBOL
A	(15,4) SIMPLEX	$X^4 + X + 1$	○
B	(15,5) BALANCED SIMPLEX (GREEN-SANSOUCIE CODE)	$X^5 + X^4 + X^2 + 1$	○
C	(15,6)	$X^6 + X^5 + 1$	△
D	(15,7)	$X^7 + X^5 + X + 1$	△
E	(15,5) INTERLACE	$X^5 + 1$	□

FIG. 1 MULTIPLE-BURST CORRECTION CAPABILITY OF REPRESENTATIVE CODES OF LENGTH 15

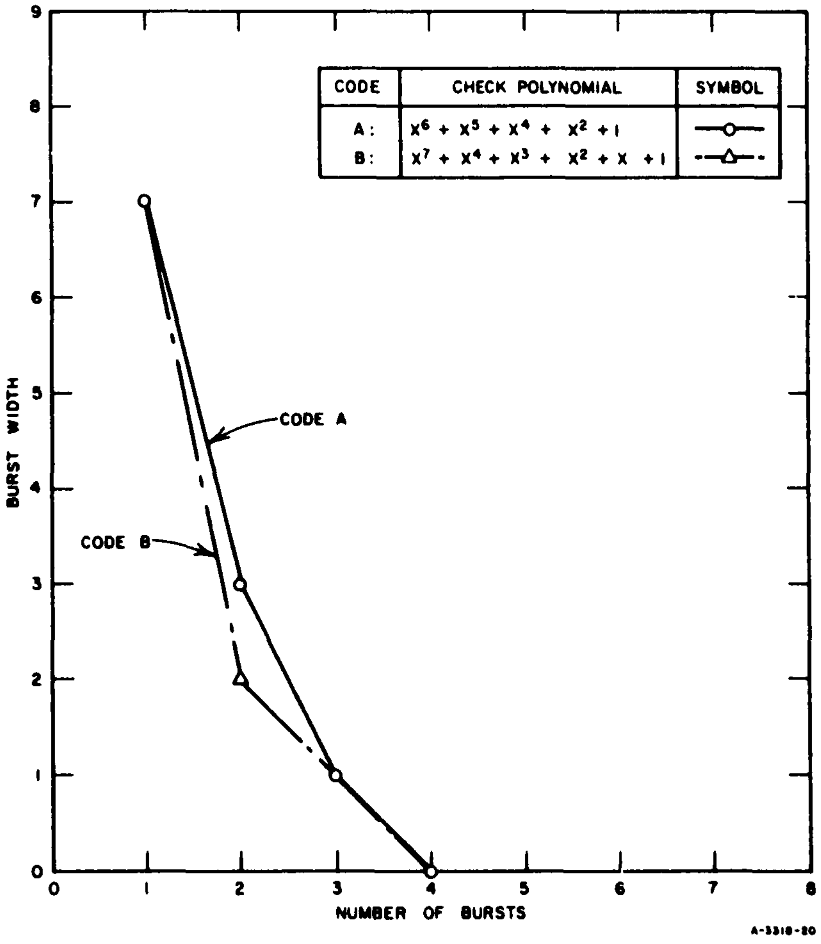


FIG. 2 MULTIPLE-BURST CORRECTION CAPABILITY OF REPRESENTATIVE CODES OF LENGTH 21

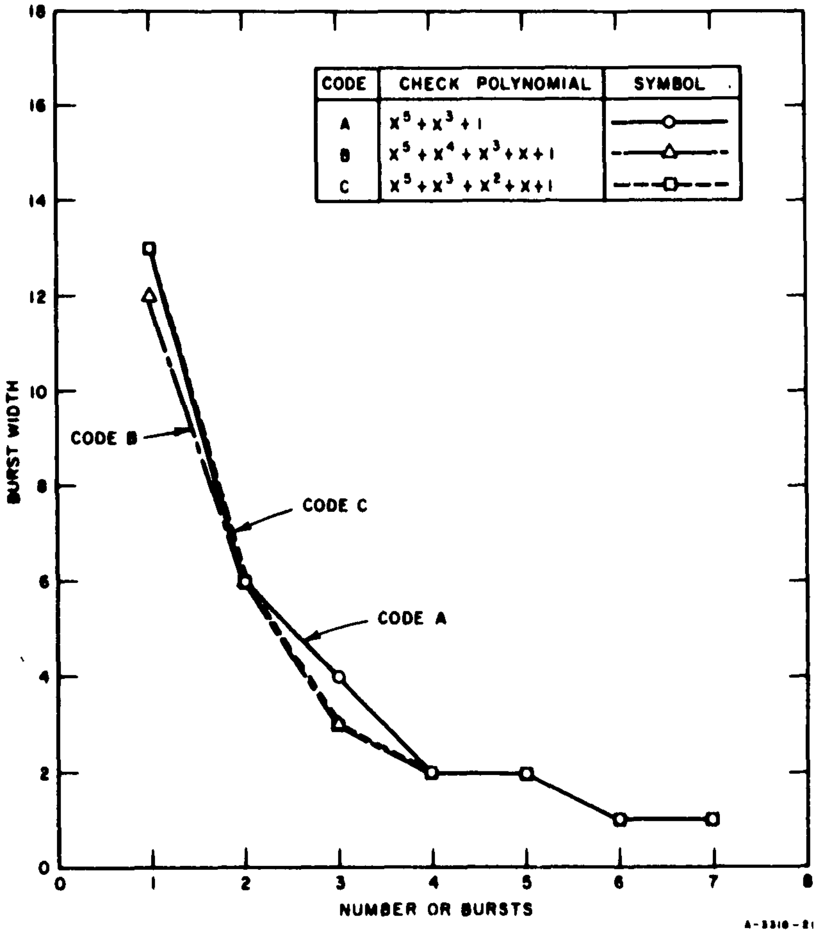


FIG. 3 MULTIPLE-BURST CORRECTION CAPABILITY OF SIMPLEX CODES OF LENGTH 31

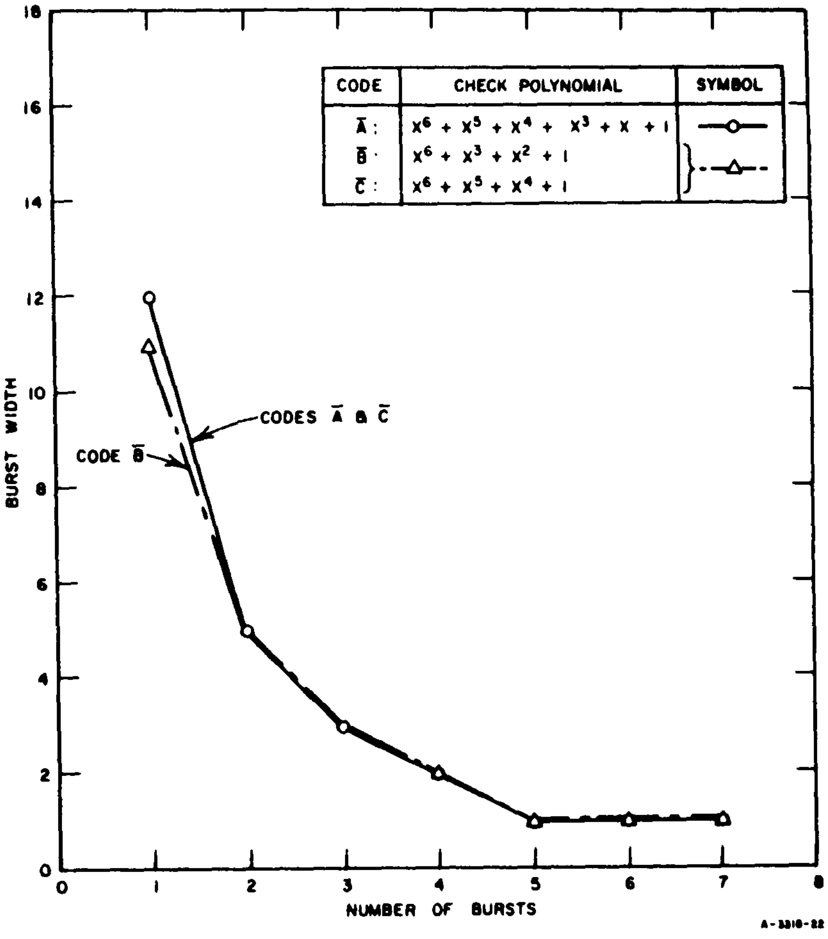


FIG. 4 MULTIPLE-BURST CORRECTION CAPABILITY OF BALANCED-SIMPLEX CODES OF LENGTH 31

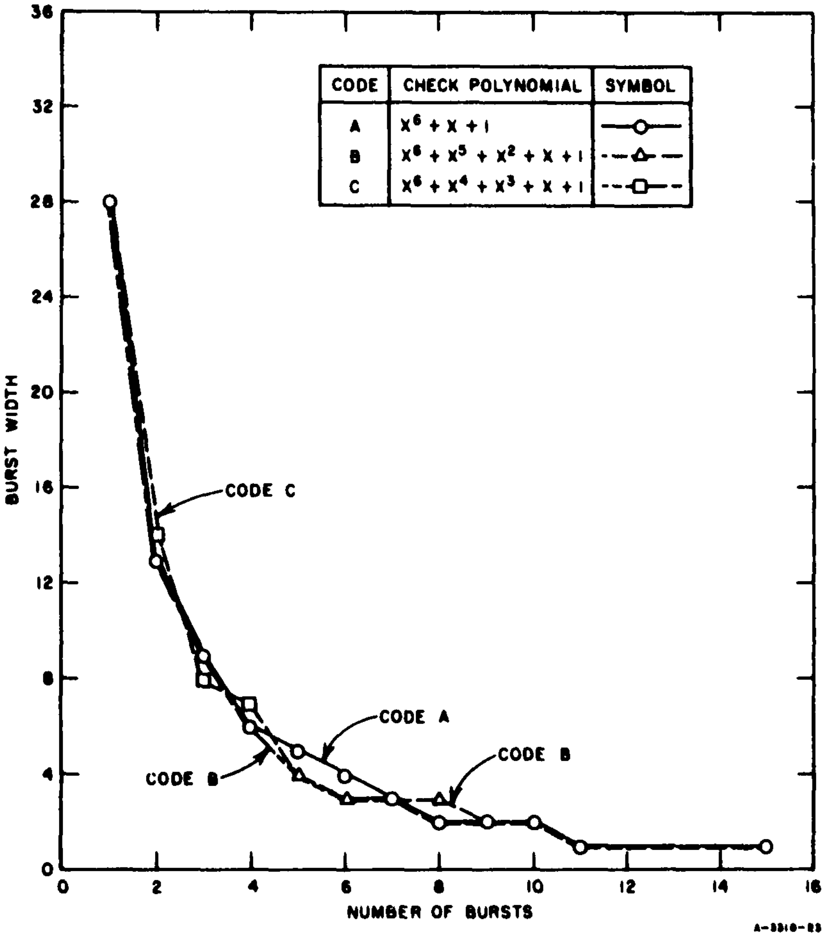


FIG. 5 MULTIPLE-BURST CORRECTION CAPABILITY OF SIMPLEX CODES OF LENGTH 63

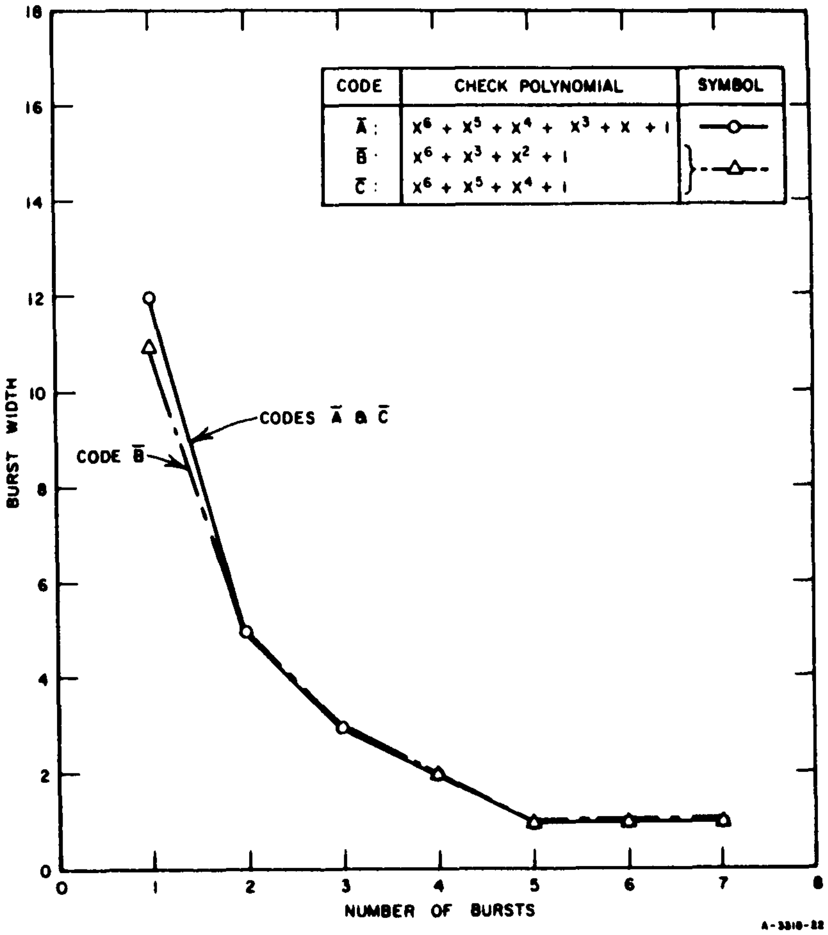


FIG. 4 MULTIPLE-BURST CORRECTION CAPABILITY OF BALANCED-SIMPLEX CODES OF LENGTH 31

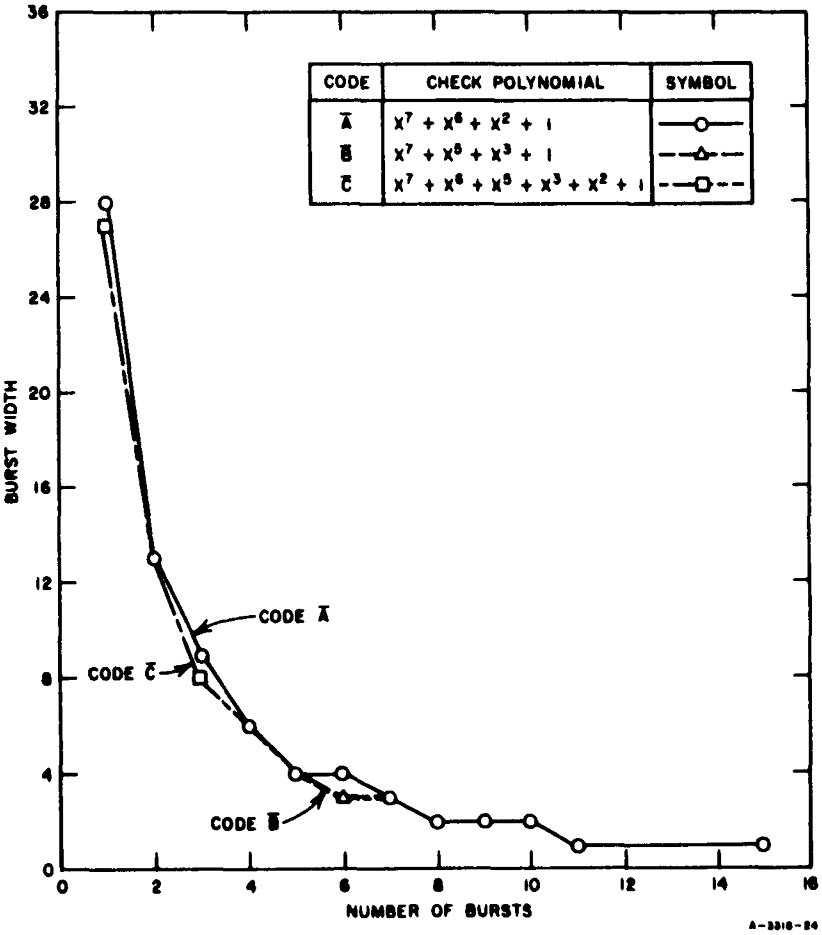


FIG. 6 MULTIPLE-BURST CORRECTION CAPABILITY OF BALANCED-SIMPLEX CODES OF LENGTH 63

IV MULTIDIMENSIONAL BURST ERROR CORRECTION

A. INTRODUCTION

Ordinary (one-dimensional) cyclic codes are useful for the correction of clustered errors, *i.e.*, bursts of errors. One might guess then that some sort of two-dimensional analog of cyclic codes would be useful for the correction of two-dimensional bursts of errors, *i.e.* "spots." This section indicates that such is indeed the case, and moreover provides a precise result for the size of a (rectangular) spot that can be corrected.

Possible applications of such codes are:

(1) Serial-parallel transmission of data over multiplexed channels subject to correlated noise bursts

(2) Multi-track magnetic tape recording

(3) High-density optical (photographic) recording of digital data

(4) Component failures in microelectronic planar arrays, etc.

The subsequent discussion will be given in terms of two-dimensional codes, but all of the results obtained are valid in the general, multidimensional case with only the obvious changes.

B. PRODUCT OF TWO CODES

Let C_1 be a block code with k_1 data bits and r_1 check bits, and with word length $n_1 = k_1 + r_1$. Let code C_2 have corresponding parameters, k_2, r_2, and $n_2 = k_2 + r_2$. We define the product code $C_1 \times C_2$ to be the (n_1n_2, k_1k_2) code described by Fig. 7.

The rows of the $k_1 \times k_2$ block, D, are data words of code C_1. Each row has appended to it r_1 redundant digits in accordance with the check rules of code C_1. These k_2r_1 check bits form the row checks, R_1. Likewise, the columns of D are regarded as data words of code C_2, and each column has appended to it r_2 check bits according to code C_2. These k_1r_2 check bits form the column checks, R_2. The block labeled R_{12} of size $r_1 \times r_2$ in the lower right-hand corner consists of checks on checks. Its digits may be formed either by taking C_1-type checks on the

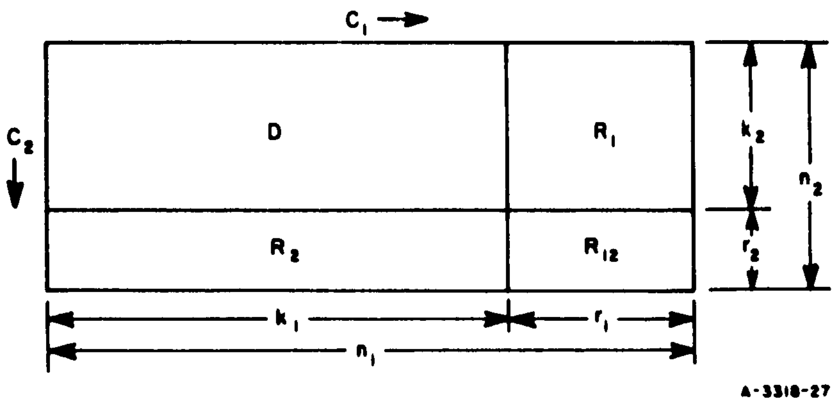


FIG. 7 PRODUCT OF TWO CODES

rows of R_2, or by taking C_2-type checks on the columns of R_1. The results are easily shown to be the same in either case.

This way of combining two codes was originally introduced by Elias[18] and called the *iteration* of codes. Slepian[19] later discussed it in a somewhat different framework as the *tensor product* of codes, showing that it is related to the idea of a tensor product of the individual code generator matrices. (See also Peterson[20]). The most important single fact about code products is the theorem (Elias,[18] Slepian,[19] Kautz[21]).

Theorem 1—If code C_i has minimum Hamming distance d_i then the product code $\prod_i C_i$ has minimum distance = $\prod_i d_i$.

Proof: See Peterson,[20] Theorem 5.3.

Other papers on multidimensional codes that have appeared are by Calingaert[22] and Rubinoff.[23]

C. PRODUCT OF TWO CYCLIC CODES

As stated at the beginning, our interest is in the product of cyclic codes, and their burst-correction capability. If C_1 and C_2 are the cyclic codes generated by polynomials $g_1(x)$ and $g_2(y)$ respectively, one would guess that their product code, $C_1 \times C_2$ would be somehow related to the product polynomial, $g_1(x)g_2(y)$. This conjecture is substantiated by the following:

Theorem 2—Let $g_1(x)$ and $g_2(y)$ belong to periods n_1 and n_2 respectively, and let C_1 be the (n_1, k_1) cyclic code generated by g_1, and C_2 be the (n_2, k_2) cyclic code generated by g_2. For any $n_2 \times n_1$ array of binary digits, (f_{ij}), let

$$f(x, y) = \sum_i \sum_j f_{ij} x^i y^j \; ; \qquad \begin{array}{l} i = 0, \ldots, n_1 - 1 \\ j = 0, \ldots, n_2 - 1 \end{array} .$$

Then (f_{ij}) is a code word in $C_1 \times C_2$ if and only if $g_1(x)g_2(y)$ divides $f(x, y)$.

Proof: Suppose first that (f_{ij}) is a code word in $C_1 \times C_2$. By definition of $C_1 \times C_2$ this means that each row (say the j_0-th row) of (f_{ij}),

$$\{(f_{ij_0}) : i = 0, \ldots, n_1 - 1\} \; ; \; j_0 = 0, \ldots, n_2 - 1$$

is a code word in C_1, and that each column (say the i_0-th column) of (f_{ij}),

$$\{(f_{i_0 j}) : j = 0, \ldots, n_2 - 1\} \; ; \; i_0 = 0, \ldots, n_1 - 1$$

is a code word in C_2. Hence the polynomials,

$$\sum_i f_{ij_0} x^i = u_{j_0}(x) \; ; \quad j_0 = 0, \ldots, n_2 - 1$$

are all divisible by $g_1(x)$, and the polynomials,

$$\sum_j f_{i_0 j} y^j = v_{i_0}(y) \; ; \quad i_0 = 0, \ldots, n_1 - 1$$

are all divisible by $g_2(y)$. It follows that

$$f(x, y) = \sum_j y^j u_j(x) = \sum_i x^i v_i(y)$$

is divisible by $g_1(x)$and also by $g_2(y)$. This in turn requires that $f(x, y)$ is divisible by the product $g_1(x)g_2(y)$.

Conversely, suppose that $f(x, y)$ is divisible by $g_1(x)g_2(y)$,

$$f(x, y) = A(x, y)g_1(x)g_2(y) = \sum_j y^j u_j(x) \quad .$$

Putting $y = 0$ in this identity,

$$f(x, 0) = A(x, 0)g_2(0)g_1(x) = u_0(x) \quad .$$

Hence $u_0(x) = \sum_i f_{i0}x^i$ is divisible by $g_1(x)$. Thus the first row of (f_{ij}) is in code C_1. Next take partial derivatives with respect to y and put $y = 0$,

$$\left.\frac{\partial f(x,\ y)}{\partial y}\right|_{y=0} = \frac{\partial}{\partial y}\left[A(x,y)g_2(y)\right]_{y=0} g_1(x) = u_1(x) \quad .$$

Hence, $u_1(x)$is divisible by $g_1(x)$. Continuing this process of taking partial derivatives with respect to y and putting $y = 0$, one sees that each row polynomial, $u_j(x)$, is divisible by $g_1(x)$. That is, each row of (f_{ij}) is in code C_1. The same reasoning with rows and columns, x and y, i and j, etc., interchanged shows that each column is in code C_2. This is clearly the same as saying that (f_{ij}) is in the code $C_1 \times C_2$. This completes the demonstration.

D. SPOT CORRECTION PROPERTIES

We first prove some easy lemmas.

> *Lemma 1*—If code C_1 has burst-correction capability b_1, then any spot (two-dimensional burst) of errors involving at most b_1 adjacent columns, is correctible in $C_1 \times C_2$.

Proof: Simply use the b_1-burst correction capability separately for each of the rows of the received code block, (f_{ij}).

Comments: Lemma 1 says that any spot, not more than b_1 bits wide, and of arbitrary height, is correctible. No use is made of the properties of code C_2; it may in fact be irredundant, $g_2(y) = 1$. By symmetry, if code C_2 is capable of correcting any burst up to b_2 bits wide, then (regardless of C_1), the product code $C_1 \times C_2$ is capable of correcting any rectangular burst not more than b_2 bits in height, and of any width.

> *Lemma 2*—If C_1 and C_2 are respectively, burst-b_1 and burst-b_2 correcting, then $C_1 \times C_2$ is capable of correcting any rectangular error spot from the set consisting of all rectangles up to b_1 bits wide, and all rectangles up to b_2 bit high, inclusive.

Proof: If the error spot consists of up to b_1 columns, use C_1 on each row; if the spot consists of up to b_2 rows, use C_2 on each column. The problem is to identify which type (vertical or horizontal) spot actually obtains in a received message. Since one or the other must obtain (by hypothesis), we may use both row checks and column checks first merely for error detection purposes to see if more than b_1 columns (or more than b_2 rows) contain errors, and act accordingly.

> *Lemma 3*—(Erasure-burst correction with cyclic code in one dimension). A cyclic code using r check bits in a block of length n can correct any erasure burst not more than r bits wide. Some erasure bursts of width $r + 1$ will not be correctible.

Proof: By cyclicity, the r erased bits may be assumed to be located in the check digit positions. But the correct digits in these locations are uniquely determined by parity checks on the remaining (data) digits, none of which have been erased. Thus the erased digits may be filled in uniquely. On the other hand, a burst of $r + 1$ consecutive erasures must involve one data digit. Thus two valid code words will agree with the received word on the unerased positions, and correction is not possible.

Lemma 3 will now be used to establish the following theorem on spot-correction capability, the principal result of this section.

> *Theorem 3*—If C_1 and C_2 are cyclic codes using r_1 and r_2 check digits respectively, with $r_1 > 0$, $r_2 > 0$, then the product code $C_1 \times C_2$ is capable of correcting any rectangular spot of dimensions r_1 bits wide by r_2 bits high, or smaller. Some spot of size $r_1 + 1$ by r_2 (or of size r_1 by $r_2 + 1$) is uncorrectible.

Proof: The method of proof involves showing first that the spot rectangle can be located within the over-all block of n_1 by n_2 digits. Next Lemma 3 is used to determine the actual pattern of errors within that rectangle. Thus the method of proof itself outlines the correction process.

First, apply all r_1 row checks to each of the n_2 rows of the received code block, purely for error detection, and note which rows contain errors. Do the same for each column, and note which columns contain errors. On the assumption that all errors are within some r_1 by r_2 block, this process will certainly locate all rows and all columns containing errors, since undetectable error patterns in a row must be more than r_1 wide, and likewise for columns. This locates a rectangle at most r_1 bits wide and at most r_2 bits high, within which all erroneous bits lie. We may now disregard completely all the received digits within this block, that is, assume they were erased. Next make use of the horizontal (or row) checks of C_1 to reconstruct the missing digits (at most r_1) in each erroneous row. This reconstitutes the correct message on the assumption that the error spot was no larger than r_1 by r_2 . Alternatively, the last step could be accomplished by using the column checks of C_2 on each erroneous column. The results should be the same, again on the same assumption. In practice, both checks may be carried out; a disagreement between the results indicates the occurrence of an error spot larger than r_1 by r_2.

To show that some error spot of size $r_1 + 1$ by r_2 is uncorrectible, consider a spot formed of r_2 identical rows, each row having errors where the polynomial $g_1(x)$ has unity coefficients. The row checks will all be satisfied in each row. The column checks will indicate errors in the appropriate columns, but we will have no information as to the rows where errors lie. Thus the actual spot is defined by the polynomial, $g_1(x)(1 + y + y^2 + \dots + y^{r_2 - 1})$, while the spot defined by $g_1(x)[g_2(y) + 1 + y + \dots + y^{r_2 - 1}]$ is also not more than $1 + r_1$ by r_2 digits in size, and it is easily seen to yield the same vertical check patterns and the same horizontal check patterns as the actual one. Thus these two spots are indistinguishable in the code $C_1 \times C_2$. Hence, not all such spots are correctible. Identical reasoning applies to spots of size r_1 by $r_2 + 1$.

> *Corollary 1*—Let V_{b_1} and H_{b_2} be the sets of rectangles, up to b_1 bits wide and up to b_2 bits high, respectively, that were the subject of Lemma 2. Let $R(r_1, r_2)$ be the set of all rectangles of height $\leq r_2$ and width $\leq r_1$ discussed in Theorem 3. We assert that $C_1 \times C_2$ is capable of correcting any spot error from the union of these classes, $V_{b_1} \cup H_{b_2} \cup R(r_1, r_2)$.

Proof: As in the proof of Theorem 3, use the row- and column-burst-detection capabilities to determine which class the error spot belongs to. Then apply the appropriate correction procedure.

According to the corollary we can correct some error rectangles wider than r_1 (or higher than r_2) provided the orthogonal dimension is restricted to be at most b_i. This does not exhaust the list of correctible error classes, as shown below.

E. EXCESS ERROR-CORRECTION CAPABILITY

It is not difficult to see that an erasure burst is correctible with a (one-dimensional) cyclic code C_g if and only if no polynomial $E(x)$ having zero coefficients on the "sure" (unerased) digit positions is a multiple of $g(x)$, except the all-zeros polynomial. This statement may be formalized as follows:

For a given erasure pattern, let $F(x)$ be the polynomial with unity coefficients on the erased positions, and zero coefficients on the "sure" positions. Another polynomial, $E(x)$, with coefficients zero or one, will be said to be a *sub-polynomial* of $F(x)$ if $f_i = 0$ implies $e_i = 0$; we then write $E(x) < F(x)$. Thus

(a) $E(x) < F(x)$

(b) $f_i = 0$ implies $e_i = 0$

(c) $e_i = 1$ implies $f_i = 1$

are all equivalent.

Lemma 4—The erasure burst represented by $F(x)$ is correctible by means of the cyclic code generated by $g(x)$ if and only if:

$$E(x) < F(x) \quad \text{and} \quad g(x) | E(x) \Rightarrow E(x) = 0 \ .$$

We note, in particular, that if $F(x) = x^c$ times a polynomial of degree less than $r = \deg(g)$, then F meets the conditions of Lemma 4, and is therefore correctible. However, there will in general exist other F's representing erasure bursts wider than r bits that are also correctible. Their existence may be exploited to broaden the class of spot error bursts correctible with a product code $C_1 \times C_2$.

Theorem 4—Let $F(x, y) = \sum_i \sum_j f_{ij} x^i y^j$ be the received error polynomial under the product code generated by $g_1(x) g_2(y)$. Also let $F_j^{(1)}(x) = \sum_i f_{ij} x^i$, and $F_i^{(2)}(y) = \sum_j f_{ij} y^j$ be the row and column vectors of the matrix f_{ij}. Define

$$S_1(x) = \bigcup_j F_j^{(1)}(x)$$

$$S_2(y) = \bigcup_i F_i^{(2)}(y)$$

to be the row- and column-logical-sum vectors, respectively. If $S_1(x)$ and $S_2(y)$ satisfy the conditions:

(1) $E(x) < S_1(x)$ and $g_1(x) | E(x) \Rightarrow E = 0$, *and*

(2) $E(y) < S_2(y)$ and $g_2(y) | E(y) \Rightarrow E = 0$,

then the error $F(x, y)$ is correctible.

Proof: Under the hypothesis of the theorem, each row vector, $F_j^{(1)}(x) < S_1(x)$; hence either $F_j^{(1)}(x)$ is the zero polynomial, or it is not divisible by $g_1(x)$. In either event, the row error vector $F_j^{(1)}(x)$ is detectable in Code C_1 (the row code). Similarly, each column error vector, $F_i^{(1)}(y)$ must represent a detectable error in C_2. We conclude that each row or column containing an error will be detected as such by the row- and column parity checks. Thus $S_1(x)$ will have a unity coefficient for each column containing any errors, and $S_2(y)$ will have a unity coefficient for each row containing any errors. We may now imagine the received digit, V_{ij}, to be erased if the term x^iy^j appears (with unity coefficient) in $S_1(x)S_2(y)$. This will result in erasure of every conceivably erroneous digit (as well as some correct ones). By virtue of the conditions (1) and (2) on S_1 and S_2 and Lemma 4, the erasure patterns thus created are correctible, either row-wise (by C_1), or column-wise (by C_2). Thus the error pattern $F(x, y)$ is correctible. Q.E.D.

One notes that, in particular, the correctibility of all rectangular bursts $r_1 \times r_2$ in size (Theorem 3) follows from Theorem 4, since if $F(x, y)$ represents a pattern of size $r_1 \times r_2$ (or less), then $S_1(x)$ is a polynomial of degree less than r_1 (times some power of x), and $S_2(y)$ is a polynomial of degree less than r_2 (times a power of y). Consequently, $g_1(x)$(whose degree is r_1) cannot divide any nonzero sub-polynomial $E(x)$ of $S_1(x)$, and $g_2(y)$ cannot divide any nonzero sub-polynomial $E(y)$ of $S_2(y)$.

Another set of special cases which come under Theorem 4 is described in the

> *Corollary* 2—If an error burst $F(x, y)$ occurs which is either
> (a) at most r_2 bits high, and is such that computation of the vertical parity checks indicates detected errors in a set of columns defined by polynomial $D_1(x)$ satisfying Condition (1) of Theorem 4, with $D_1(x)$ replacing $S_1(x)$, or
> (b) at most r_1 bits wide, and is such that computation of the horizontal parity checks indicates detected errors in a set of rows defined by polynomial $D_2(y)$ satisfying Condition (2) of Theorem 4, with $D_2(y)$ replacing $S_2(y)$,
> then $F(x, y)$ is correctible with the code $C_1 \times C_2$.

The set of errors covered by this Corollary is of special interest in that it is defined in part by a test on the outcome of the receiver's parity testing, *i.e.*, by an implicit condition.

F. CORRECTION OF ERASURE BURSTS

We imagine that information encoded in $C_1 \times C_2$ is transmitted over a binary erasure channel. Thus the received word, (v_{ij}), is a matrix of

0's, 1's, and X's (erased digits), where the 0's and 1's are assumed to be free of errors. As usual in the erasure situation, we have the advantage of knowing precisely which digits were erased. Thus no error location is involved—only a filling-in of the blanks (X's).

For a received message, (V_{ij}), we define

$$f_{ij} = 0\ , \quad \text{when} \quad v_{ij} = 0 \quad \text{or} \quad 1$$

$$f_{ij} = 1\ , \quad \text{when} \quad v_{ij} = X \quad .$$

The polynomial $F(x, y) = \sum_i \sum_j f_{ij} x^i y^j$ will be called the *erasure polynomial*. $F(x, y)$ thus gives the locations of the erased digits of the message.

Theorem 5 below gives a necessary and sufficient condition on $F(x, y)$ for correctibility of an erasure pattern with code $C_1 \times C_2$ generated by $g_1(x)g_2(y)$.

> *Theorem* 5—If $F(x, y)$ is such that no nonzero subpolynomial of F is a multiple of $g_1(x)g_2(y)$, then the code $C_1 \times C_2$ generated by $g_1 g_2$ will correct erasure burst $F(x, y)$ uniquely. Conversely, if there does exist a nonzero polynomial $E(x, y) < F(x, y)$ with $E(x, y) = a(x, y)g_1(x)g_2(y)$, then there are at least two distinct code words of $C_1 \times C_2$ that agree with the received word on the nonerased positions, and hence the received word is not correctible in this code.

Proof: Take the second half first—if $0 \neq E(x, y) < F(x, y)$ and $E(x, y) = a(x, y)g_1(x)g_2(y)$, let $U(x, y)$ be the transmitted word. Then U and V differ only on the erased digit positions [*i.e.*, on $F(x, y)$]. Hence U and $U + E$ differ only on the erased digit positions. But $U(x, y) = b(x, y)g_1(x)g_2(y)$ since U is in the code $C_1 \times C_2$. Hence $U + E = (a + b)g_1 g_2$, so that $U + E$ is also in the code. Thus U and $U + E$ are two valid code words differing only on the erased digit positions. The received word is then not uniquely correctible. Conversely, suppose that no nonzero subpolynomial of F is a multiple of $g_1 g_2$. If there were two distinct code words, say, $U = bg_1 g_2$ and $U' = b' g_1 g_2$ such that $U + U'$ is nonzero only on the erased digit positions, then $U + U' = (b + b')g_1 g_2$ is a nonzero multiple of $g_1 g_2$ and is also a subpolynomial of F, *i.e.*, $U + U' < F$ contrary to hypothesis.

> *Corollary 3*—Code $C_1 \times C_2$ is capable of correcting any erasure burst that leaves a rectangular block of $k_1 \times k_2$ unerased digits,

where $k_1 = n_1 - r_1$ = number of information digits in the row code, and $k_2 = n_2 - r_2$ = number of information digits in the column code.

This result also follows directly from the observation that any k_1 consecutive digits in a cyclic (n_1, k_1) code determine the remaining r_1 digits uniquely. Hence, all the erased digits may be filled in uniquely.

V ERROR--LOCATING CODES

A. INTRODUCTION

By way of introduction to the concept of error location, let us consider how error-detecting codes and error-correcting codes, respectively, are used in communication systems. Error-correcting codes may be employed on one-way communication systems, since the crucial operations of detecting the presence of errors, locating the specific symbols in error, and finally correcting these erroneous symbols may all be carried out at the receiver. With error-detecting codes, the situation is quite different. Here the amount of code redundancy is sufficient only to permit the receiver to determine that errors have occurred, but not, in general, to locate or correct these errors. Consequently, if the receiver wishes to determine the correct message, a request for repeat transmission must be sent back to the transmitter. Thus, a return link must be provided.

Such two-way systems with error detection at the receiver and "decision feedback" to the transmitter have many attractive features and have come in for considerable study in recent years. They are particularly efficient in situations where the channel fails "catastrophically," that is, where channel disturbances are usually slight, but occasionally become extremely severe. The reason for this is that error-detecting codes require relatively little redundancy. A single over-all parity check digit will detect one half of the possible failure patterns, two parity checks will detect all but one-quarter of the possible failure patterns, and in general, r suitably chosen parity checks will detect all but a fraction 2^{-r} of the possible failure patterns, regardless of the block length involved. Thus, if the block length is reasonably great, the actual information rate will be very nearly the channel digit rate during periods of good transmission. When the channel becomes very noisy, on the other hand, we may suppose that its instantaneous capacity falls nearly to zero. Under such conditions, the best one can do is to give up trying until conditions improve, and this is essentially what the decision feedback system does. By way of contrast, a one-way error-correcting code system would have to employ very large amounts of redundancy to combat the noise during the poor transmission periods. The decision feedback system automatically adjusts its data rate to match (roughly) the conditions on the channel.

There is, however, a fairly crucial choice of parameters that must be made in designing a decision feedback system. In particular, the block length must be made neither too large nor too small. If the block length is small, the fractional redundancy required to detect most of the likely error patterns will be large, leading to a low data rate, even under good transmission conditions. If the block length is chosen too large, it becomes increasingly likely that nearly every received block will contain some detected errors. Since every block detected to be in error must be repeated, the data rate again falls to zero. Incidentally, this argument also shows why decision feedback with error detection at the receiver is not a good scheme for channels where the error rate is fairly high all the time. For such channels, one may be caught on the horns of a dilemma in trying to choose a suitable value of block length.

Fortunately, there appears to be at least one solution* to this difficulty, which preserves the other attractive features of the decision feedback system. A wasteful aspect of the error-detection system described thus far is that the presence of even a single erroneous digit in a received message requires the repeat of a whole block of data, and we would like to make this block long for reasons of code efficiency. Might it not be possible to employ a long code block with relatively low redundancy, but with the redundancy so arranged that when errors (of some restricted class) occur we can determine roughly which portions of the over-all block are erroneous? The remainder of this section discusses several techniques for the construction of such *error-locating* codes, that is, codes that permit the receiver to determine which portion or portions of the over-all code block are in error. The feedback message in such a system then signals the transmitter to repeat only the erroneous portions of the code block. The additional flexibility thus afforded in system design may then be exploited to soften the compromise between short and long block lengths.

*Suggested by J. Wolf, (private communication).

From a certain point of view, these error-locating codes may be regarded as lying between error-detecting codes and error-correcting codes both in their capabilities and in the amount of redundancy required. Whereas error detection answers only the question, "Have any errors occurred?", and error correction answers the question, "Precisely which digits are in error, if any?", error location answers the question, "Which portions of the received message are in error?". We shall refer to such error-locating codes as *EL codes*.

B. A SIMPLE EXAMPLE OF AN EL CODE

We give here a simple example of an EL code to illustrate the basic ideas involved. The parameters (length, redundancy, etc.) for this particular code are not the most attractive ones that can be exhibited for such codes, since in the interests of clarity of exposition, they have deliberately been chosen to be rather small. In later paragraphs of this section, we shall give examples of more useful codes, as well as several classes of construction techniques for these codes.

Let A be the check matrix (of size 2 × 3) for the ordinary Hamming single-error-correcting (SEC) code of length 3 binary digits.

$$A = \begin{bmatrix} 1 & 0 & 1 \\ 0 & 1 & 1 \end{bmatrix} .$$

This matrix has the property that the modulo-two sum of any two (or fewer) columns is nonzero, that is, the (3, 1) code defined by A is a two-error-detecting code. In fact, the sum of all three columns is zero; hence the sum of any two columns is itself the remaining column of A.

Now let us use A as a basic structural unit to build up a larger matrix, H, which will be the check matrix for the desired EL code. Let H be the partitioned matrix,

$$H = \begin{bmatrix} A & 0 & A & A & A \\ 0 & A & A & A' & A'' \end{bmatrix}$$

where

$$A' = \begin{bmatrix} 1 & 1 & 0 \\ 1 & 0 & 1 \end{bmatrix} \quad \text{and} \quad A'' = \begin{bmatrix} 0 & 1 & 1 \\ 1 & 1 & 0 \end{bmatrix}$$

Thus

$$H = \begin{bmatrix} 101 & 000 & 101 & 101 & 101 \\ 011 & 000 & 011 & 011 & 011 \\ \\ 000 & 101 & 101 & 110 & 011 \\ 000 & 011 & 011 & 101 & 110 \end{bmatrix}$$

This matrix H defines a binary group code of length $n = 15$, employing 4 check digits. The 15 places of the code are considered as consisting of 5 *sub-blocks*, each having 3 places. Note that the matrices A' and A'' are obtained from A by column permutation; hence, they have the same basic property as A, given above.

Now let us consider what properties are possesed by the (15, 11) code defined by H. First of all, the 15 columns of H are exactly the 15 distinct binary vectors of length 4. Hence, this code is in one sense simply a peculiar version of the Hamming (15, 11) SEC code. However, the particular way in which the columns are grouped into sub-blocks of length 3 makes the following properties hold.

(1) The (modulo-two) sum of any two (or fewer) columns of H located in the *same sub-block* is nonzero.

(2) The (modulo-two) sum of any two (or fewer) columns of H drawn from the same sub-block is distinct from any such sum formed from a *different* sub-block.

Property (1) follows directly from the stated property for matrix A. Property (2) is a little more difficult to justify, but note first that sub-blocks 1 and 2 are uniquely characterized by the fact that one-fold or two-fold column sums drawn from them have the forms,

$$\begin{bmatrix} a \\ b \\ 0 \\ 0 \end{bmatrix} \quad \text{and} \quad \begin{bmatrix} 0 \\ 0 \\ a \\ b \end{bmatrix}$$

respectively. Column sums (of one or two columns) drawn from sub-block 3 have the form,

$$\begin{bmatrix} a \\ b \\ a \\ b \end{bmatrix}$$

where a and b are not both zero. On the other hand, the corresponding forms derived from sub-blocks 4 and 5 are, respectively:

$$\begin{bmatrix} a \\ b \\ a + b \\ a \end{bmatrix} \quad \text{and} \quad \begin{bmatrix} a \\ b \\ b \\ a + b \end{bmatrix} \quad .$$

It is clear that 4-place vectors of these forms can agree only if $a = b = 0$, which is ruled out by the property of A.

Now the syndrome* resulting from the occurrence of one or two digit errors within a given sub-block of the code is precisely the sum of the columns of the check matrix H corresponding to these erroneous digit positions. It follows [by Property (2)], that the syndrome resulting from any such error pattern will uniquely determine the particular sub-block (1, 2, 3, 4 or 5) in which the error(s) occurred. And, of course, this syndrome will be nonzero [by Property (1)] thus distinguishing it from the syndrome for the case of no errors. The (15, 11) code defined by H is therefore capable of locating a single sub-block containing one or two digit errors. We shall refer to it as a 2-EL code.

Note, on the other hand, that the occurrence of three errors in any given sub-block is undetectable, since it leads to the all-zeros syndrome. In general, the occurrence of errors in more than one sub-block will lead to incorrect decoding (that is, the code will not locate more than one erroneous sub-block).

As mentioned earlier, this illustrative example of an EL code is not a particularly attractive code. The same code could be used to *correct* any single digit error, or alternatively, to *detect* the presence of any two digit errors anywhere in the full block of fiteen digits. Further, using it as an EL code, we can also locate the offending sub-block of three digits, provided that all errors occurred in a single sub-block. We shall see that more striking comparisons can be made for longer codes where EL codes show up in a more favorable light.

*See Reference 20, p. 36.

C. GENERAL EL CODES

1. Terminology and Notation

We shall employ the symbols, n, r and k in their customary senses. Thus,

n = over-all block length (code word length)

r = number of redundant digits (check digits) per word

$k = n - r$ = number of information digits per word.

The symbols, t, s and e will be used as follows:

t = length of a sub-block

s = number of sub-blocks per word

e = maximum number of digits that may be in error within a given sub-block.

It will be assumed that all sub-blocks have the same length, t. Hence $n = st$. A code with the above parameters will be referred to as an *e-EL* $(n, n\text{-}r)$ *code of sub-block length* t. Only codes for the location of a single erroneous sub-block will be considered here. (It is still an open question as to whether there exist efficient codes for the location of more than one erroneous sub-block.)

2. Basic Properties

We formalize here the two basic properties (already discussed for the illustrative code in Sec. V-B) in the general case. Let H be the check matrix for an e-EL $(n, n\text{-}r)$ code of sub-block length t (see Fig. 8). H is a matrix of r rows and n columns, the columns being grouped into s sub-blocks, each of length t.

Let C^a_j stand for the j-th column within sub-block a of the matrix H. Then we may represent an arbitrary sum of e or fewer columns of H, all drawn from sub-block a, by the expression:

$$\sum_{i=1}^{\leq e} C^a_{j_i}$$

where the integers, j_1, j_2, ..., j_e represent an arbitrary set of e column positions.

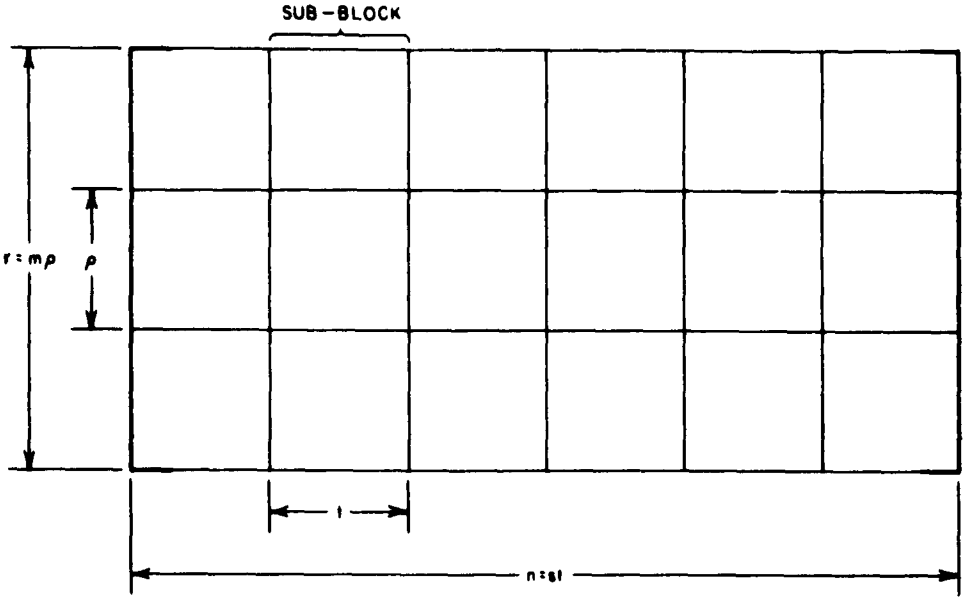


FIG. 8 CHECK MATRIX STRUCTURE FOR ERROR-LOCATING CODES

The two necessary and sufficient conditions for *H* to define an *e*-EL code may then be written:

(a) $\sum_{i=1}^{\leq e} C^{a}_{j_i} \neq 0$, for any (non-empty) set $\{j_i\}$.

(b) If $a \neq b$, then $\sum_{i=1}^{\leq e} C^{a}_{j_i} \neq \sum_{i=1}^{\leq e} C^{b}_{k_i}$ for any (non-empty) sets $\{j_i\}$ and $\{k_i\}$.

The summation signs here refer, of course, to digit-wise modulo-two summation of the *r*-place column vectors of *H*. The two expressions appearing in Conditions (a) and (b) therefore represent syndromes resulting from *e* (or fewer) errors in sub-block *a* and sub-block *b* respectively. These syndromes must be distinct and nonzero if the code is to be e-error locating, and conversely.

It is not difficult to show that the following bounding relation holds for any EL code:

$$1 + s \sum_{i=1}^{[\frac{t}{2}]} \binom{t}{i} \leq 2^{r} \qquad (1)$$

Proof: We count the distinct syndromes. Since, within a given sub-block all f-fold sums of columns $(f \leq e)$ are nonzero by Condition (a), it follows that within a given sub-block all h-fold sums of columns ($h \leq [e/2]$*) must be distinct. Moreover, f-fold sums of columns from different sub-blocks are distinct by Condition (b) $(f \leq e)$. Hence, all h-fold sums of columns are distinct when up to $[e/2]$ columns at a time are chosen from any one of the s sub-blocks. There are

$$1 + s \sum_{i=1}^{[e/2]} \binom{t}{i}$$

ways of making such choices, including the zeros syndrome. But there are only 2^r distinct r-place binary vectors. Hence the result.

Any EL code meeting this bound with equality will be said to be an optimum EL code. The particular code described in Sec. V-B was an optimum EL code.

3. A Family of Optimum 2-EL Codes

A family of 2-EL codes generalizing the code of Sec. V-B is readily described. All the codes of this family are optimum EL codes.

Let A be the check matrix of ρ rows and t columns for the Hamming SEC code of length $t = 2^\rho - 1$. Let $A', A'', \ldots, A^{(t-1)}$ be the matrices formed by cyclically permuting the columns of A. The check matrix H_2 for the 2-EL code of size $n = 2^{2\rho} - 1$ is formed according to the following scheme:

$$H_2 = \begin{bmatrix} A & 0 & A & A & A & \cdots & A \\ 0 & A & A & A' & A'' & \cdots & A^{(t-1)} \end{bmatrix}.$$

We have $s = t + 2 = 2^\rho + 1$, $n = st = (2^\rho + 1)(2^\rho - 1) = 2^{2\rho} - 1$, and $r = 2\rho$.

The proof that this code is indeed capable of locating any single sub-block containing up to 2 errors is given in Appendix I (Theorem 1). The code obviously meets the bound, Eq. (1).

A further generalization is obtained by using the scheme:

* $[x]$ = the largest integer $\leq x$.

$$H_3 = \left[\begin{matrix} A & 0 & 0 & A & A & \dots & A & A & A & A & \dots & A \\ 0 & A & 0 & A & A' & \dots & A^{(t-1)} & 0 & 0 & 0 & \dots & 0 \\ 0 & 0 & A & 0 & 0 & \dots & 0 & A & A' & A'' & \dots & A^{(t-1)} \end{matrix}\right.$$

$$\begin{matrix} 0 & 0 & 0 & \dots & 0 & A & A & A & \dots & A \\ A & A & A & \dots & A & A' & A' & A' & \dots & A' \\ A & A' & A'' & \dots & A^{(t-1)} & A & A' & A'' & \dots & A^{(t-1)} \end{matrix}$$

$$\left.\begin{matrix} \dots\dots\dots & A & A & \dots & A \\ \dots\dots\dots & A^{(t-1)} & A^{(t-1)} & \dots & A^{(t-1)} \\ \dots\dots\dots & A & A' & \dots & A^{(t-1)} \end{matrix}\right]$$

which contains $3 + 3t + t^2 = [(t + 1)^3 - 1]/t$ sub-blocks each of width t. Thus for this family of codes, $r = 3\rho$ and $t = 2^{\rho} - 1$, and $n = 2^{3\rho} - 1$. The proof that such codes are two-error locating is given in Appendix I, Theorem 2.

The obvious generalization of schemes H_2 and H_3 to the case H_m ($m \geq 2$) yields 2-EL codes of length $n = 2^{m\rho} - 1$, $r = m\rho$, and containing sub-blocks of length $t = 2^{\rho} - 1$. The check matrix H_m consists of m sets of rows, each of ρ rows. The sub-blocks fall into groups, the i-th group containing those sub-blocks where i of the row sets ($i = 1, \dots, m$) are nonzero, while $m - i$ of the row sets are zeros. Within each group all $\binom{m}{i}$ choices of $m - i$ zero row sets are used, and for each such choice all t^{i-1} permutations of the lower $i-1$ row matrices, $A^{(j)}$, relative to the uppermost nonzero row set are employed.

This procedure thus yields a total of $m + \binom{m}{2}t + \binom{m}{3}t^2 + \dots + \binom{m}{m}t^{m-1} = [(1 + t)^m - 1]/t$ sub-blocks, that is, a total of $(1 + t)^m - 1 = n$ digits per code word. See also Appendix I. Again these codes are optimum 2-EL codes.

4. A Family of (t-1)-EL Codes

The (15, 11) 2-EL code of Sec. V-B can also be generalized in the direction of providing detection of a larger number of errors per sub-block. In fact, the codes of this family will locate any sub-block (length t) that contains up to $t - 1$ errors. Thus, these codes will detect almost all the possible errors in a single sub-block. Surprisingly, the cost in redundancy of providing this error location ability is not high.

The parameters of this family are:

t = any prime ≥ 3

$e = \rho = t - 1$

$r = m\rho$, where m = any integer ≥ 2,

$n = (t + 1)^m - 1$.

For example, using $m = 2$ one has the codes with

$t = 5$, $e = 4$, $r = 8$, $n = 35$

$t = 7$, $e = 6$, $r = 12$, $n = 63$, etc.

The second of the above codes, with $t = 7$, stands up very well in comparison with the $n = 63$, $r = 12$ Bose-Chaudhuri code, which is capable of detecting any 4 digits in error out of a block of 63. The EL code, having the same redundancy and same block length is capable of locating any sub-block containing up to 6 errors.

The basic check matrix, A, for the construction of EL codes of the $e = t - 1$ family is:

$$A = \begin{bmatrix} 1 & 0 & 0 & 0 & \ldots & 1 \\ 0 & 1 & 0 & 0 & \ldots & 1 \\ 0 & 0 & 1 & 0 & \ldots & 1 \\ 0 & 0 & 0 & 1 & \ldots & 1 \\ . & . & . & . & . \; . \; . & . \\ 0 & 0 & 0 & 0 & \ldots 1 & 1 \end{bmatrix} .$$

In order to form the check matrix H for these codes, one uses the basic matrix A in the schemes, H_2, H_3, or generally, H_m, which have already been discussed above. For example,

$$H = \begin{bmatrix} 10001 & 00000 & 10001 & 10001 & 10001 & 10001 & 10001 \\ 01001 & 00000 & 01001 & 01001 & 01001 & 01001 & 01001 \\ 00101 & 00000 & 00101 & 00101 & 00101 & 00101 & 00101 \\ 00011 & 00000 & 00011 & 00011 & 00011 & 00011 & 00011 \\ & & & & & & \\ 00000 & 10001 & 10001 & 00011 & 00110 & 01100 & 11000 \\ 00000 & 01001 & 01001 & 10010 & 00101 & 01010 & 10100 \\ 00000 & 00101 & 00101 & 01010 & 10100 & 01001 & 10010 \\ 00000 & 00011 & 00011 & 00110 & 01100 & 11000 & 10001 \end{bmatrix}$$

is the over-all check matrix for the 4-EL code with $t = 5$, $m = 2$, $c = t - 1 = 4$, $n = t^2 + 2t = 35$, $s = 7$.

Unfortunately, the codes of this family, $e = t - 1$, are not optimum EL codes for $e > 2$. For example, application of the bound inequality for the case $t = 5$ given above, yields that $s_{opt} = (2^r - 1)/(5 + 10) = 255/15 = 17$, instead of 7, and that correspondingly, $n_{opt} = (5)(17) = 85$ instead of merely 35.

The proof that the codes of this family are indeed e-EL codes, with $e = t - 1$ depends on a somewhat more general theorem which we state here, and whose proof is given in Appendix I.

> *Theorem 1*—Let $g(x) = g_1(x)g_2(x)\ldots\ g_\nu(x)$ be a product of ν *distinct*, irreducible polynomials over $GF(2)$, where each irreducible factor $g_i(x)$ belongs to the same period, t. That is, t is the smallest integer such that $g_i(x)$ divides $x^t + 1$ (mod 2). Let A be an ρ-row-by-t-column check matrix for the cyclic $(t,\ t - \rho)$ code C_g, then the check matrix formed by substituting A into the scheme H_m yields an e-EL code of size $n = (t + 1)^m - 1$, with $r = m\rho$ check digits, where $e + 1 = d$ is the Hamming distance of the code C_g, and $\rho = deg(g)$ is the number of check digit positions in code C_g.

The codes of the family $e = t - 1$ are obtained from Theorem 1 when one chooses$g(x) = x^{t-1} + x^{t-2} + \ldots + x + 1 = (x^t + 1)/(x + 1)$, where t is an odd prime. The oddness of t guarantees that $g(x)$ will have no repeated factors, and the primeness of t guarantees that all the irreducible factors of $g(x)$ have the same period. Thus one may choose $t = 3, 5, 7, 11, 13, 17, \ldots$ for this family of codes.

5. Other EL Codes

One may also use Theorem 1 with polynomials $g(x)$ other than the all-ones polynomial which gave rise to the $e = t - 1$ family. For example, choosing $g(x)$ to be a primitive, irreducible polynomial of degree ρ, and period $t = 2^\rho - 1$ will yield the EL codes of the family $e = 2$ described in Sec. V-C-3. If $g(x)$ is chosen to be an irreducible, but not primitive polynomial, *e.g.*, $x^8 + x^5 + x^4 + x^3 + 1$, with $t = 17$, then one still obtains an EL code. In this case, $d = 5$, hence the code is capable of locating a sub-block (width $t = 17$) containing up to 4 errors and (using $m = 2$) the block length $n = 323$, there are $r = 16$ check digits, and $s = 19$ sub-blocks.

The Golay code polynomial may be used in similar fashion. Here $g(x) = x^{11} + x^9 + x^7 + x^6 + x^5 + 1$, generating a cyclic code of length $t = 23$ and distance 7. (This happens to be a perfect code.) Again using $m = 2$ in the construction, the resulting EL code has $n = (23)(25) = 575$, $r = 22$ and permits the location of a sub-block containing up to 6 errors.

VI MAVERIC--A VERSATILE ENCODER-DECODER FOR CYCLIC CODES

A. GENERAL DESCRIPTION

The name MAVERIC stands for MAgnetic VERsatile Information Corrector. This device, which was developed as part of the present study, is an extremely versatile encoder-decoder for cyclic error-detecting and error-correcting codes. The adjective "magnetic" is used because all of the storage functions and practically all of the logic in the machine are provided by multiaperture magnetic cores.

MAVERIC is completely self-contained; the only external connection required is to an ac power source, 105-130 volts at 60 cycles per second. Its power consumption varies from about 34 watts (standby) to 130 watts (maximum). The clock rate is set internally to about 500 bits per second.

The major subassemblies of MAVERIC are:

(1) 63-stage message register

(2) 16-stage feedback shift register

(3) Twelve 16-stage binary sequence sensors

(4) Cycle counter

(5) Prime driver and pulse amplifier unit

(6) Pulse driver

(7) Control unit

(8) Power Supply

(9) Associated switches and indicators

(10) Plugboard for programming codes.

These subassemblies are interconnected through the plugboard and switch wiring so as to provide maximum versatility in use. More than 65,000 different codes may be set up. A mode selector switch reconfigures the circuit to operate either as an encoder or as an error-correcting decoder, or simply as a feedback shift register with provision for stopping the clock at an arbitrarily preselected state (or states). This last mode (called the SPECIAL mode) is useful in connection with the study of the properties of feedback shift register sequences that have application to spread-spectrum communication techniques, and to radar, as well as to error-correcting codes.

B. ENCODE MODE OPERATION

In the ENCODE mode of operation, MAVERIC is configured to operate as an encoder for cyclic codes. The code generating polynomial is selected by means of a bank of 16 toggle switches that control the feedback taps on a 16-stage feedback shift register. The code word length, *n*, the number of information digits, *k*, and the number of check digits, $r = n - k$, are selected by plugboard programming.

The last *n* stages of the message resister (where $n \leq 63$) function in this mode as a circulating register to hold the information digits and check digits of a code word. Arbitrary information digits may be entered into the last *k* stages of this register by means of individual toggle switches for each stage. The data is set into the corresponding cores when the ENTER DATA button is pressed. When the ADVANCE button is pushed, the information digits are circulated back through the *n*-stage register, and are simultaneously applied to the *r*-stage feedback shift register. After exactly *k* clock pulses, the output of this feedback shift register is applied to the input of the *n*-stage register and appropriate check digits follow the circulated information digits into the *n*-stage register. The whole operation consumes *n* clock times, at the end of which time a valid encoded word appears in the *n*-stage register.

C. DECODE MODE OPERATION

Decoding of a received code word occupies two full word times (*i.e.*, 2*n* clock times). The first *n* clock times are used to perform *r* parity check calculations, which in effect compare the *r* received digits in the check digit positions with recomputed check digits obtained from the received information digits. The result is an *r*-place binary number, which appears in the syndrome register (*r*-stage feedback shift register). A binary *zero* here represents agreement, a binary *one* represents disagreement. This *r*-place binary number is usually called a *syndrome* (or check word).

During the next *n* clock times, this syndrome is converted to a signal that performs the actual correction of the received word. Only the syndrome calculation (first *n* clock times) is performed during DECODE mode operation. The actual correction operation is carried out with the selector switch in CORRECTION mode position. (This operation is described below.)

In DECODE mode the machine is configured so as to circulate the entire *n*-bit word in the message register, and also so as to apply this whole word to the *r*-bit syndrome register. The resulting syndrome appears as the contents of the syndrome register at the end of this (*n*-bit) cycle. If the syndrome register contains all zeros at this time, this is an indication that the word in the message register is a valid code word, and no corrections are necessary. Thus the execution of the DECODE mode constitutes an error detection check on the received word.

Errors may be deliberately introduced into a received word by means of separate ERROR SWITCHES associated with the message register. It is convenient to use these instead of the ENTER DATA toggles since the error switches may then serve as a mechanical register, which remembers the error locations. The indicator lights associated with the message register may be used to indicate either actual data or disagreements between actual data and the ENTER DATA SWITCH positions by means of a two-position register switch with NORMAL and ERROR positions.

D. CORRECTION MODE OPERATION

This mode executes the second half of the decoding cycle, wherein the syndrome is converted to a correction signal applied to the received message.

The message stored in the *n*-stage register is circulated, one bit at a time, while the syndrome register is advanced, one clock at a time, in the absence of inputs from the message register. Meanwhile, a bank of sequence sensors are monitoring the output of the feedback bus, which drives the syndrome register input. Whenever an *r*-bit sequence occurs that has been set up for detection by a sequence sensor, a correction signal is given to alter the bit that happens to be circulating around the end of the message register. In principle, a separate sequence sensor is required for each error pattern to be corrected (*e.g.*, a single error, *x*, or a double-adjacent error, *xx*, or the error pattern *xox*, etc.). However, in the case of burst-error correcting codes, which correct all error patterns up to a given width *b*, a single sequence sensor making use of "don't care" conditions will suffice. In that case, the sequence sensor looks for $r-b$ consecutive binary zeros.

Since a maximum of twelve separate sequence sensors is available on MAVERIC, it is clear that the programming of multiple-error-correcting codes (other than burst error correction codes) will be somewhat limited. The number of sequence sensors required for double-error correction in code words of length *n* is given by:

$$S_2 = 1 + \frac{n-1}{2} = \frac{n+1}{2}$$

since there are $(n-1)/2$ cyclically distinct configurations of two errors out of *n*, in general. (Obviously, only one sequence sensor is required for single-error correction, *i.e.*, $S_1 = 1$.) Similarly, the number of sequence sensors required for triple-error correction is

$$S_3 = 1 + \frac{n-1}{2} + \frac{(n-1)(n-2)}{6} = \frac{n^2+5}{6} .$$

Similar expressions are readily obtained for higher-order codes.

The above expressions indicate that MAVERIC is limited to two-error-correcting codes with $n \leq 23$, and to three-error-correcting codes with $n \leq 8$. There do exist a variety of two-error correcting codes of this size (including the famous (23, 12) code of Golay, used as a two-error-correcting code). Moreover, it turns out that a number of the three-error-correcting codes, when they are used only for the correction of two random errors, are also capable of correcting a sizeable number of longer burst patterns.

E. SPECIAL MODE OPERATION

The fourth position of the mode selector switch provides a mode of operation of MAVERIC, which is not directly related to encoding or decoding operations with error-correcting codes. This SPECIAL mode permits the feedback shift register to be operated independently of the message register, and moreover makes provision for the automatic stopping of the master clock under control of the output of sequence sensors Nos. 1-4.

This mode may be used for the following purposes:

(1) To determine the cycle length (period) of a given generating polynomial (up to degree 16),

(2) To determine the length of a given cycle in the cycle set of a given generating polynomial,

(3) To determine whether two feedback shift register states are on the same or different cycles,

(4) To generate continuously a pseudo-noise sequence.

In counting cycle lengths (as in 1 and 2 above), it is necessary to connect a pulse counter (scaler) to the clock output jack on MAVERIC. The longest cycle of any given generating polynomial is obtained by starting the feedback register in the state, (1000...0), and setting up a sequence sensor to look for the sequence, 000...01.

In order to determine whether two register states belong to the same or different cycles, one starts the register in one of these two states and sets up a sequence sensor (from among Nos. 1-4) to stop the clock when the other state is reached. An attached pulse counter gives the relative displacement of the two states, assuming that they fall on the same cycle.

The pseudo-noise sequence corresponding to any given generating polynomial up to degree 16, and to any given initial state, may be sampled at the feedback bus output jack.

F. LOGIC CIRCUIT AND COMPONENTS

As mentioned above, all of the storage and logic functions performed in MAVERIC are accomplished through the use of multiaperture magnetic cores (using the so-called MAD-R technique[24]). In this category are:

(1) The 63-bit message register,

(2) The 16-bit feedback register,

(3) The twelve 16-bit sequence sensors, and

(4) The cycle counter.

These components are interconnected by means of the plugboard and magnetic gating circuits (under control of the program switches). In some instances, transistors and silicon-controlled rectifiers are used to amplify signals to the magnetic logic where large fan-out is required.

A simplified equivalent logic diagram for MAVERIC is shown in Fig. 9. The control inputs marked "Encode", "Decode" and "Correct" are energized by means of the selector switch, depending on the function desired. The logical gate circuits, energized by one or another of these three control inputs, determine the mode of interconnection of the registers, cycle counter and sequence sensors.

It is of interest to compare the numbers of basic circuit components (diodes, resistors, capacitors, MADs, etc.) used in MAVERIC with the numbers which would be required in an equivalent transistor machine

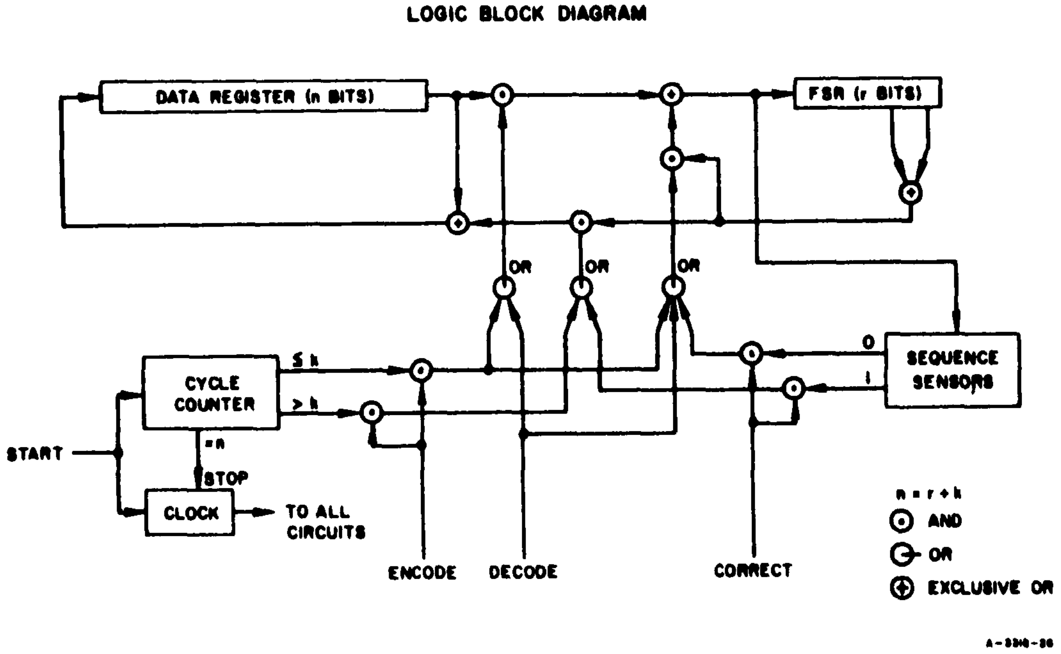


FIG. 9 LOGIC BLOCK DIAGRAM FOR MAVERIC

(not using MAD logic). A rough—but realistic—estimate has been made for such a transistor version, more with reliability considerations in mind than comparative cost. The results are shown in Table III.

Table III

COMPONENT COUNT OF MAVERIC

	Message Register	FSR and Logic	Sequence Detectors	Counter	Prime Drivers	Pulse Gener-ators	Control	Mag-netics Driver	Total	Equiva-lent Tran-sistor Machine
Transistors and SCRS	--	--	--	--	4	5	4	4	17	220
Diodes	--	--	--	--	--	--	--	4	4	458
Resistors	--	--	--	--	8	15	11	15	49	848
Capacitors	--	--	—	--	5	7	4	4	20	384
MADs	126	36	388	37	--	--	--	--	587	--
Toroids	--	16	4	6	1	1	2	10	40	--
Solder Joints	395	75	1600	75	50	74	57	74	2400	4300

Note: Indicator circuits and dc power supply not included. Solder joints at patch board and front panel not included.

It should be noted that the indicator circuits and dc power supply have not been included in this comparison (for either version) since the numbers and types of components would be closely comparable for the two versions. Moreover, the solder joints at the patch board and at the front panel have also been ommitted from the table for the same reason.

Solder joints have been included as a basic circuit component because with the increasing reliability of electronic components, the solder joint has become a significant factor in determining overall reliability. It will be observed that the actual (magnetic logic) MAVERIC contains far fewer solder joints (2400 as compared to 4300) than would an equivalent transistor machine. Comparison of the numbers of diodes, resistors, and capacitors are even more striking, as a glance at Table III will show.

The total number of magnetic components (MADs and simple ferrite toroids) in MAVERIC is 627 as compared to about 220 three-terminal semiconductor devices (transistors and SCRs) in the transistor version.

However, the basic reliability of magnetic components has been estimated to be from 10 to 100 times better (in terms of mean failure rate under normal use) than the reliability of semiconductors. Thus the magnetic components in MAVERIC still have a considerable edge in over-all reliability over the semiconductors in a transistor machine. For some environments—radiation environments, for example—the advantage of magnetic components would be still more pronounced.

The above conclusions must, of course be tempered with the realization that the operating speeds of magnetic components and semiconductors are not comparable. Magnetic components of the type used in MAVERIC are limited to bit rates of several kilocycles at best, while transistors readily operate inthe megacycle range. Nevertheless, for applications where high operating speeds are not required, the higher reliabilities attained with magnetic components make them much more desirable than conventional semiconductor components.

It may be noted that our experience with MAVERIC during the debugging stages and early operating stages definitely tended to confirm the above conclusions. All of the troubles encountered were traced to failure of transistors or silicon controlled rectifiers, principally in the indicator and power supply circuits. In no case, was failure attributable to faulty cores or core wiring.

APPENDIX I

PROOFS OF THEOREMS ON ERROR-LOCATING CODES

APPENDIX I

PROOFS OF THEOREMS ON ERROR-LOCATING CODES

1. EL CODES GENERATED BY A SINGLE IRREDUCIBLE POLYNOMIAL

The following theorem covers the family of 2-EL codes described in Sec. V-C-3.

> *Theorem 1*—Let $g(x)$ be a primitive irreducible polynomial of degree ρ over $GF(2)$, and let α be a root of $g(x) = 0$ in the field $GF(2^{p})$ Form the matrix A,
>
> $$A = [1, \alpha, \alpha^2, \ldots, \alpha^{t-1}] \quad ; \quad t = 2^{\rho} - 1 \qquad \text{(I-1)}$$
>
> where each entry α^{j} is to be thought of as a vector (ρ-tuple) in $V_{\rho}[GF(2)]$. The matrix H_2
>
> $$H_2 = \begin{bmatrix} A & 0 & A & A & A & . & . & . & A \\ 0 & A & A & \alpha A & \alpha^2 A & . & . & . & \alpha^{t-1}A \end{bmatrix} \qquad \text{(I-2)}$$
>
> is then the check matrix for an $(n, n-r)$ code capable of locating any single sub-block of length t that contains up to two digit errors (a 2-EL code) where $n = t(t + 2)$ and $r = 2\rho$.

Proof: As defined, A is the check matrix for a cyclic Hamming code of distance 3 and length $t = 2^{\rho} -1$. Hence the sum of any two (or fewer) columns of A, $\alpha^{i} + \alpha^{j}$, is nonzero. Therefore the sum of any two (or fewer) columns from the same sub-block of H_2 is likewise nonzero. This proves Condition (a) (see Sec. V-C-2) for the EL code, with $e = 2$.

It remains to show that sums of two (or fewer) columns of H_2 drawn from sub-block a cannot be equal to any sum of two (or fewer) columns of H_2 drawn from a different sub-block b. [See Condition (b), Sec. V-C-2]. The first two sub-blocks are clearly distinguishable both from each other and any other sub-blocks by virtue of the weight property of A already used. Column sums drawn from sub-blocks a and b of the form $\begin{matrix}A\\ \alpha^{a}A\end{matrix}$ and $\begin{matrix}A\\ \alpha^{b}A\end{matrix}$ may be written as

$$\sum_i \begin{pmatrix} \alpha^{a+j_i} \\ \alpha^{a+j_i} \end{pmatrix} \quad \text{and} \quad \sum_i \begin{pmatrix} \alpha^{k_i} \\ \alpha^{b+k_i} \end{pmatrix}$$

respectively. If these sums are to be equal, then

$$\sum_i \alpha^{j_i} = \sum_i \alpha^{k_i} \neq 0 \tag{I-3}$$

and

$$\sum_i \alpha^{a+j_i} = \sum_i \alpha^{b+k_i} \tag{I-4}$$

By dividing the Eq. (I-4) of these relations by Eq. (I-3), we obtain

$$\alpha^a = \alpha^b$$

$$\alpha^{a-b} = 1 , \qquad \text{where} \qquad 0 \leq a,\ b < t = 2^{\rho} - 1$$

Hence

$$a = b$$

That is, the sub-blocks a and b are the same sub-block.

Since A has ρ rows and t columns, H_2 will have, in all, 2ρ rows and $t(t + 2)$ columns.

The proof of Theorem 1 actually demonstrates more than has been asserted, since if $g(x)$ is an irreducible, but not necessarily a *primitive* polynomial, which generates a cyclic code C_g of length t [= period of $g(x)$] and distance d, then H_2 will define an EL code detecting up to $e = d-1$ errors within one sub-block. We thus have the

> *Corollary*—If $g(x)$ is an irreducible polynomial of degree ρ over $GF(2)$, and α is a root of $g(x) = 0$, then the matrix
>
> $$A = [1, \quad \alpha, \quad \alpha^2, \quad . \quad . \quad . \quad , \quad \alpha^{t-1}]$$
>
> (where $\alpha^t = 1$, and the α^j are all distinct for $j = 0,\ 1,\ \ldots,\ t - 1$), is the check matrix for a Bose-Chaudhuri code of some distance d. The EL code defined by

$$H_2 = \begin{bmatrix} A & O & A & A & . & . & . & A \\ O & A & A & \alpha A & . & & . & \alpha^{t-1}A \end{bmatrix}$$

is then capable of locating a single sub-block which contains up to $e = d - 1$ errors. The block length $n = t(t + 2)$ and the number of check digits is 2ρ.

In Sec. V-C-3, there was described also a family of 2-EL codes involving a parameter $m = 2, 3, \ldots$ where $r = m\rho$ and $n = (t + 1)^m - 1$. A slight extension to Theorem 1 (and its corollary) suffices to demonstrate the capabilities of these codes. We give the details here for the case $m = 3$, and where $g(x)$ is not necessarily primitive. The appropriate modifications for the case of arbitrary m will then be obvious.

Theorem 2—Let $g(x)$ be an irreducible polynomial over $GF(2)$ of degree ρ and period t. Let A be the check matrix for the cyclic $(t, t - \rho)$ code of distance d generated by $g(x)$. Let H_m be the check matrix formed from A by the method of Sec. V-C-3. Then H_m defines an e-EL code of length $n = (t + 1)^m - 1$, with $r = m\rho$, and $e = d - 1$.

Proof: (for the case $m = 3$). The check matrix A has the form, $A = [1, \alpha, \alpha^2, \ldots, \alpha^{t-1}]$. The check matrix H_3 is obtained by multiplying by A each of the entries of the matrix:

$$\left[\begin{array}{ccccccccccccccccccccc} 1 & 0 & 0 & 1 & 1 & \ldots & 1 & 1 & 1 & \ldots & 1 & 1 & 1 & \ldots & 1 & 1 & 1 & \ldots & 1 \\ 0 & 1 & 0 & 1 & \alpha & \ldots & \alpha^{t-1} & 0 & 0 & \ldots & 0 & 1 & 1 & \ldots & 1 & \alpha & \alpha & \ldots & \alpha \\ 0 & 0 & 1 & 0 & 0 & \ldots & 0 & 1 & \alpha & \ldots & \alpha^{t-1} & 1 & \alpha & \ldots & \alpha^{t-1} & 1 & \alpha & \ldots & \alpha^{t-1} \end{array}\right.$$

$$\left.\begin{array}{ccccccccc} 1 & 1 & \ldots & 1 & \ldots & 1 & 1 & \ldots & 1 \\ \alpha^2 & \alpha^2 & \ldots & \alpha^2 & \ldots & \alpha^{t-1} & \alpha^{t-1} & \ldots & \alpha^{t-1} \\ 1 & \alpha & \ldots & \alpha^{t-1} & \ldots & 1 & \alpha & \ldots & \alpha^{t-1} \end{array}\right]$$

The resulting matrix clearly has $t^2 + 3t + 3 = [(t + 1)^3 - 1]/t$ columns and 3 rows (3ρ rows after the quantities α^i are expanded into binary form). By virtue of the assumed distance, d, for the code generated by A, it follows that no sum of fewer than d columns chosen from any one sub-block of H_3 can be zero. It remains to be shown that such sums determine uniquely the sub-block from which they are chosen. This is immediately obvious for those sub-blocks of H_3 containing one or more rows of zeros. The remaining sub-blocks have columns of the form:

$$\begin{pmatrix} \alpha^{i} \\ \alpha^{a+i} \\ \alpha^{b+i} \end{pmatrix}$$

Any identity of column sums chosen from different sub-blocks will then imply a relation of the form:

$$\sum_{k=1}^{\leq d-1} \begin{pmatrix} \alpha^{i_k} \\ \alpha^{a+i_k} \\ \alpha^{b+i_k} \end{pmatrix} = \sum_{k=1}^{\leq d-1} \begin{pmatrix} \alpha^{j_k} \\ \alpha^{c+j_k} \\ \alpha^{d+j_k} \end{pmatrix} \tag{I-5}$$

From this relation one concludes immediately that

$$\Sigma\, \alpha^{i_k} = \Sigma\, \alpha^{j_k} \neq 0 \tag{I-6}$$

$$\alpha^{a}\, \Sigma\, \alpha^{i_k} = \alpha^{c}\, \Sigma\, \alpha^{j_k} \tag{I-7}$$

$$\alpha^{b}\, \Sigma\, \alpha^{i_k} = \alpha^{d}\, \Sigma\, \alpha^{j_k} \tag{I-8}$$

By dividing Eq. (I-7) and Eq. (I-8) by Eq. (I-6), one has

$$\alpha^{a} = \alpha^{c}$$

and

$$\alpha^{b} = \alpha^{d}$$

Since 1, α, ..., α^{t-1} are all distinct and since $0 \leq a, b, c, d < t$, it follows that $a = c$ and $b = d$; hence the two sub-blocks from which identical column sums were formed are in fact the same sub-block. Consequently, the EL code defined by H_3 will locate up to $d - 1$ errors in one sub-block.

2. EL CODES GENERATED BY A PRODUCT OF IRREDUCIBLE POLYNOMIALS

We outline here the proof of Theorem 1 of Sec. V-C-4.

Let $g(x) = g_1(x)g_2(x) \ldots g_\nu(x)$ be a product of ν

distinct irreducible polynomials over $GF(2)$, where each factor g_i belongs to the same period t_i. Let A be the check matrix for the cyclic $(t, t - \rho)$ code, C_g, generated by $g(x)$. Then the check matrix formed by substituting A into the scheme H_m (see Sec. V-C-3) defines an e-EL code of size $n = (t + 1)^m - 1$, with $r = m\rho$ check digits, where $e + 1 = d$ is the Hamming distance of the code C_g, and $\rho = \deg(g)$ is the number of check digit positions in code C_g.

Proof: The assumed distance property of code C_g guarantees that no sum of fewer than d columns of H_m chosen from the same sub-block can vanish. For the same reason, such column sums chosen from sub-blocks with one or more vanishing rows will be characteristic of those sub-blocks. The only difficult portion of the proof concerns those sub-blocks all of whose rows are nonzero. Let

$$A = \begin{bmatrix} 1 & \alpha_1 & \alpha_1^2 & \cdots & \alpha_1^{t-1} \\ 1 & \alpha_2 & \alpha_2^2 & \cdots & \alpha_2^{t-1} \\ \cdot & \cdot & \cdot & \cdots & \cdot \\ 1 & \alpha_\nu & \alpha_\nu^2 & \cdots & \alpha_\nu^{t-1} \end{bmatrix} \tag{I-9}$$

be the check matrix for C_g, where α_i is a root of $gi(x) = 0$ for $i = 1, \ldots, \nu$. (Observe that A is a Vandermonde matrix.)

Substitution of A into the scheme H_m must here be understood in terms of cyclic permutation of the columns of A (rather than multiplication by powers of α) since $g(x)$ contains more than one irreducible factor. The assumption of an identity between column sums (of fewer than d columns) from distinct sub-blocks of H_m implies the relations:

$$\sum_{k=1}^{<d} \alpha_i^{i_k} = \sum_{k=1}^{<d} \alpha_i^{j_k} \tag{I-10}$$

$$\sum_{k=1}^{<d} \alpha_i^{i_k+a} = \sum_{k=1}^{<d} \alpha_i^{j_k+c} \tag{I-11}$$

$$\sum_{k=1}^{<d} \alpha_i^{i_k+b} = \sum_{k=1}^{<d} \alpha_i^{j_k+d} \tag{I-12}$$

etc.

The distance property of C_g implies that Eq. (I-10) is nonzero for at least one value of $i = 1, \ldots, \nu$. Using this value of i, divide Eqs. (I-11, 12) by Eq. (I-10). The result for this value of i is then

$$\alpha_i^a = \alpha_i^c$$

and

$$\alpha_i^b = \alpha_i^d$$

Since $1, \alpha_i, \alpha_i^2, \ldots, \alpha_i^{t-1}$ are all distinct it follows that $a = c$ and $b = d$ (and similar relations for $m > 3$). But this means that the two sub-blocks chosen were in fact the same one. Hence the code locates up to d errors in one sub-block, as asserted.

APPENDIX II

MAXIMAL SETS OF RELATIVELY PRIME POLYNOMIALS

APPENDIX II

MAXIMAL SETS OF RELATIVELY PRIME POLYNOMIALS

In constructing multiple-burst error correcting codes by means of the Chinese Remainder Theorem, as in Sec. III-D, it is usually convenient to use as moduli, polynomials of the same degree, d. These polynomials need not be irreducible—only relatively prime (in pairs)—and they must not, of course, be divisible by x.

It is therefore of interest to determine the largest set of relatively prime polynomials, of given degree d and with nonzero constant term, over the field $GF(2)$. In this appendix we determine the size, $P(d)$, of such a maximal set, $S = \{m_i(x)\}$, in terms of known arithmetic functions, and also give an algorithm for constructing maximal sets.

Let $I(d)$ = number of irreducible polynomials of degree d. It is well known* that

$$\sum_{d|n} dI(d) = 2^n \tag{II-1}$$

and

$$I(n) = \frac{1}{n} \sum_{d|n} \mu(d) 2^{n/d} \tag{II-2}$$

Moreover, $I(d)$ is a monotone increasing function of d for $d \geq 2$. For convenience, we redefine $I(1) = 1$ (thus excluding the polynomial x, and making the function monotone non-decreasing everywhere). Of course, Eqs. (II-1) and (II-2) then require minor modifications; but we shall have no further need of these equations here.

Clearly

$$P(d) \geq I(d) \tag{II-3}$$

*The Möbius function $\mu(d) = 0$ if d contains a perfect square factor; otherwise $\mu(P_1 \ldots Pr) = (-1)^r$, where the P_i are distinct primes.

since the set of dth degree irreducibles is certainly a suitable (but probably non-maximal) set.

We now determine how many additional polynomials of degree d can be constructed, all prime to each other and prime to the irreducibles. Any polynomial in such a set is either irreducible (of degree d, and hence already counted), or it contains at least one irreducible factor of degree $\leq [d/2]$. But for each integer i, $1 \leq i \leq [d/2]$, we may use each irreducible of degree i no more than once in constructing products for the set. Our set can therefore contain no more additional polynomials than there are irreducibles of degrees $i \leq [d/2]$. Hence,

$$P(d) \leq I(d) + \sum_{i=1}^{[d/2]} I(i) \tag{II-4}$$

Next we exhibit an algorithm for constructing a set meeting this upper bound exactly. First, suppose that d = odd integer. For each partition of $d = i + (d - i)$ into two parts, $i = 1, \ldots, (d - 1)/2$, we use the $I(i)$ irreducibles of degree i, each one once in association with a suitable irreducible of degree $d - i > i$. There are always enough irreducibles of degree $d - i$ available, since $I(d - i) \geq I(i)$. These products together with the $I(d)$ irreducibles of degree d, form a set of the required size. Next, if d is even, the same procedure is followed, except that the case $i = d/2$ must be considered as well. For $i = d/2$, form the squares of the $I(d/2)$ irreducibles of degree $d/2$. These are clearly prime to each other and to the previously formed products. Again the total number of polynomials formed is given by Eq. (II-4). Hence,

$$P(d) = I(d) + \sum_{i=1}^{[d/2]} I(i) \tag{II-5}$$

gives the exact size of a maximal set of relatively prime polynomials of degree d (none of which are divisible by x).

The following table is readily constructed.

n	$I(n)$	$P(n)$
1	1*	1
2	1	2
3	2	3
4	3	5
5	6	8
6	9	13
7	18	22
8	30	37
9	56	63
10	99	112
11	186	199
12	335	357
13	630	652
14	1161	1201
15	2182	2222
16	4080	4150

Observe that the excess, $P(n) - I(n)$, becomes small compared to $I(n)$ as n increases beyond about $n = 10$. Thus one does not gain much for very long codes by using the additional (reducible) polynomials.

* As redefined; strictly speaking $I(1) = 2$, including the polynomial x.

APPENDIX III

PERFORMANCE BOUND FOR MULTIPLE-BURST ERROR-CORRECTING CODES

APPENDIX III

PERFORMANCE BOUND FOR MULTIPLE-BURST ERROR-CORRECTING CODES

It is possible to derive a lower bound to the number of check digits, r, required for a code that corrects up to m bursts of errors, each of width not exceeding b digits. This bound is obtained by counting the number of distinct error configurations, $g(n,m,b)$ over a block of n binary digits.

Two cases must be distinguished, depending on whether only *open-loop* bursts or also *closed-loop* bursts are to be made correctible. The closed-loop case has proven to be completely intractable in analytic terms, so that only the open-loop formula is given here.

Once the formula for $g(n,m,b)$ has been found, one may assert that the minimum number, r, of check digits required satisfies:

$$2^r \geq g(n,m,b)$$

Several different methods* have been found for determining the function $g(n,m,b)$ explicitly. The simplest of these methods follows.

An error pattern correctible in an (open-loop) m,b code can be regarded as an n-place binary vector with the following property: It is possible to "cover" all the binary *ones* of the vector by using no more than m masks, each of width b places. The problem then is to determine the number of distinct vectors (out of a total of 2^n) that possess this property. Such vectors will be called (m,b) *coverable*.

Any n-place, (m,b) coverable vector must contain at least $n - mb$ zeros. These are the digits not covered by any mask. In addition there may be as many as $m(b - 1)$ other zero digits that are covered by masks. For sake of

* The method used here is due to M. W. Green of the Computer Techniques Laboratory. Another, more complicated, enumeration scheme was developed by J. J. Stone earlier in the project.

definiteness, we assume that the masks are placed so that the leftmost digit under any mask is a *one*. Thus, any given vector can be tested for (m,b) coverability by laying down masks according to the above convention, working from left to right. The vector is coverable if and only if no more than m masks are needed to cover all the *ones*.

The number of distinct mask placements (using j masks) possible is clearly the binomial coefficient,

$$C_j^{n-m(b-1)}$$

Since there are $n - m(b - 1)$ possible locations for the j masks. For any one mask placement, there are exactly $2^{m(b-1)}$ distinct error vectors coverable by this placement, since the $b - 1$ places (other than the leftmost position under each mask) may contain either zeros or ones. Summing over the possible values of $j = 0, 1, \ldots, m$ one obtains:

$$g(n,m,b) \quad = \quad 2^{m(b-1)} \sum_{j=0}^{m} C_j^{n-m(b-1)}$$

Note that therefore,

$$g(n,m,b) \quad = \quad 2^{m(b-1)} g[n - m(b - 1),\ m,\ 1]$$

so that the fractional number (out of 2^n vectors) that are (m,b) coverable is a function only of the parameters, $n' = n - m(b - 1)$ and m, and not explicitly of the burst width, b.

$$\text{Fraction coverable} \quad = \quad 2^{-n} g(n,m,b) \quad = \quad 2^{-n'} g(n',\ m,\ 1)$$

The resulting bound on the number of check digits required is, therefore:

$$r \geq \log_2 \left\{ \sum_{j=0}^{m} C_j^{n-m(b-1)} \right\} + m(b - 1)$$

It is not difficult to show that an asymptotic expression for $g(n,m,b)$ valid when $n \gg m$ and $n \gg b$ is given by:

$$g(n,m,b) \sim \frac{2^{m(b-1)} \; n^{m}}{m!}$$

Hence, asymptotically

$$r \geq m(b - 1) + m \log_2 n - \log_2 (m!)$$

APPENDIX IV

QUASI-CYCLIC, SHORTENED CYCLIC AND PSEUDO-CYCLIC CODES

APPENDIX IV

QUASI-CYCLIC, SHORTENED CYCLIC AND PSEUDO-CYCLIC CODES

1. QUASI-CYCLIC CODES

Much of the great utility of cyclic codes stems from the property that all of the r parity check relations satisfied by a cyclic code are of the same form, extending over the preceding $k - 1$ digits. This fact suggests that a generalization of cyclic codes preserving only this property might be useful.

Let us call any code defined by a parity check matrix of the form:

$$H = \begin{bmatrix} h_0 & h_1 & h_2 & \dots & h_{k-1} & 1 & & 0 & 0 & \dots & 0 \\ 0 & h_0 & h_1 & \dots & h_{k-2} & h_{k-1} & & 1 & 0 & \dots & 0 \\ 0 & 0 & h_0 & \dots & h_{k-3} & h_{k-2} & h_{k-1} & 1 & & \dots & 0 \\ \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot \\ 0 & 0 & 0 & \dots & & & & & & \dots & 1 \end{bmatrix} \qquad \text{(IV-1)}$$

a *quasi-cyclic code.*

Such codes may be encoded by using a suitable nonlinear feedback shift register to generate the sequence of binary digits,

$$h_0,\ h_1,\ h_2,\ \dots,\ h_{k-1},\ h_k = 1$$

for use as a gating signal employed against the incoming data digits in order to generate the check digits.

Cyclic codes constitute the special case of quasi-cyclic codes for which the block length n is such that the polynomial

$$h(x) = h_0 + h_1 x + h_2 x^2 + \dots + h_k x^k$$

divides $x^n + 1$ without remainder.

2. SHORTENED CYCLIC CODES

If C_g is a cyclic (n,k) code, let C'_g be the $(n-i,\ k-i)$ code obtained by setting i of the information places of C_g identically equal to zero. That is, C'_g is that subspace of C_g for which

$$x_1 = x_2 = \dots = x_i = 0$$

in every code word. These i digits may, of course, be dropped; the resulting code C'_g may be viewed then as a linear subspace of the $n-i$ dimensional space, $V_{n-i}(2)$ over $GF(2)$.

The check matrix for C'_g may be written as:

$$H' = \left[\;\underbrace{\begin{matrix}0&0&0&\cdots&0\\0&0&0&\cdots&0\\\cdots&\cdots&\cdots&\cdots&\cdots\\0&0&0&\cdots&0\end{matrix}}_{i}\;\middle|\;\underbrace{\begin{matrix}h'_0&h'_1&\cdots&1&0&0&\cdots&0\\0&h'_0&\cdots&\cdot&1&0&\cdots&0\\\cdots&\cdots&\cdots&\cdots&\cdots&\cdots&\cdots&\cdots\\0&0&\cdots&\cdots&\cdots&\cdots&\cdots&1\end{matrix}}_{n-i}\;\right] \qquad \text{(IV-2)}$$

where the first i columns of the matrix are identically zero and may therefore be dropped.

Obviously, one obtains in this manner exactly those check matrices of the form (IV-1) used to define quasi-cyclic codes. Hence, quasi-cyclic and shortened cyclic codes are the same thing.

3. ANOTHER CHARACTERIZATION OF QUASI-CYCLIC CODES

Quasi-cyclic codes may also be regarded in another light.* Consider the set of all polynomials,

$$f(x) = f_0 + f_1x + f_2x^2 + \dots + f_{n-1}x^{n-1} \qquad \text{(IV-3)}$$

such that for a fixed polynomial, $g(x)$, one has $f(x) = m(x)g(x)$. That is, we consider the set $C_{g,n}$ of all polynomial multiples of g whose degree is less than n.

* This viewpoint was, in fact, used by J. J. Stone as a definition of quasi-cyclic in the Interim Technical Report, Sec. V-C.

$$C_{g,n} = \{\text{set of all } f(x): \; g|f \quad \text{and} \quad \deg(f) < n\}$$

Clearly, if n happens to be the period of $g(x)$, then $C_{g,n}$ will be a cyclic code. If not, then $C_{g,n}$ is quasi-cyclic, as will now be demonstrated.

If $g(x)$ has period n' (assumed $> n$),* *i.e.*, if $g(x)|x^{n'} + 1$, then $C_{g,n'}$ is cyclic and it has the generators:

$$g(x),\ xg(x),\ x^2g(x),\ \ldots,\ x^{k'-1}g(x)$$

where $k' = n' - \deg(g) = n' - r$. We now inquire as to the code with the generators:

$$g(x),\ xg(x),\ x^2g(x),\ \ldots,\ x^{k-1}g(x)$$

where

$$k = n - r$$

which are a subset of the generators for $C_{g,n'}$. Clearly this new code is the shortened code obtained from $C_{g,n'}$ by setting zeros into the last $i = k' - k = n' - n$ places. But this shortened code is precisely $C_{g,n}$, since the above k generators span $C_{g,n}$. As already observed, shortened codes are quasi-cyclic; hence $C_{g,n}$ is a quasi-cyclic code, and all three characterizations given thus far coincide.

4. PSEUDO-CYCLIC CODES

Still another viewpoint has been provided by Peterson.[20] *Pseudo-cyclic* codes are defined as ideals in the algebra of polynomials taken modulo an arbitrary polynomial, $K(x)$. When $K(x) = x^n - 1$, one obtains cyclic codes as a special case.

It was shown by Peterson that:

(a) Every shortened cyclic code is a pseudo-cyclic code for some choice of $K(x)$, and

(b) Every pseudo-cyclic code of weight greater than two is a shortened cyclic code.

* For we may always replace n' by $2n'$, $3n'$, ... etc.

Consequently one concludes that the concepts, quasi-cyclic, pseudo-cyclic, shortened cyclic are all substantially equivalent.

5. APPLICATIONS

There are good indications[25] that in many cases it is possible to find shortened cyclic codes for (single) burst error correction which are better than any cyclic codes. In particular Kasami[25] has found such codes for the correction of error bursts of certain lengths achieving a longer block length, n, than any cyclic codes found thus far with the same values of b and r. However, the improvement is slight.

Such codes probably deserve additional study. They may be regarded from any of the viewpoints described above (quasi-cyclic, pseudo-cyclic or shortened cyclic); however, they are best implemented by regarding them as shortened cyclic codes.

APPENDIX V

RELIABILITY CONSIDERATIONS

APPENDIX V

RELIABILITY CONSIDERATIONS

1. INTRODUCTION

A question that is often asked in regard to error-correction techniques is the following:

"When one inserts redundancy into a message for the purpose of error control, more digits have to be sent to convey the same amount of information to the receiver. If, as is often the case, the actual data rate is fixed (by real-time considerations), the transmission bandwidth has to be widened to carry the increased channel bit rate. But this tends to degrade the signal-noise ratio and worsen the bit error probability (before error correction). How does one then know that there will be an over-all improvement after error correction is carried out?"

One answer to this question has been provided by Klein[26], who shows that the message reliability for coded messages will exceed that for uncoded transmission under broad assumptions regarding the functional dependence of the bit reliability $q = 1 - p$ on the normalized channel bit rate, R.

A comparable question may be asked from an equipment reliability viewpoint. The addition of redundancy to a message requires extra terminal equipment, the amount depending on the complexity and sophistication of the coding scheme used. An over-all view of the problem of reliable digital data transmission demands that the reliability of the terminal equipment be considered as well as errors introduced in the transmission channel proper. In view of this, it is reasonable to inquire whether the addition of extra terminal equipment (which may contribute errors of its own) can actually result in an over-all degradation of performance, and if so, under what circumstances.

Some thought was given to this question during the course of the project reported on here. It is the purpose of this appendix to describe the results and conclusions arrived at regarding this question.

2. FACTORS ENTERING INTO EQUIPMENT RELIABILITY

It is almost immediately apparent that any realistic discussion of equipment reliability factors will be considerably more complex than answering the channel error question posed in the preceding paragraph. Sources of channel errors, and their statistics, are fairly well understood for a wide variety of real channels. In other, less well-understood situations, one can always obtain some degree of guidance by resorting to simplifying assumptions (independent errors, gaussian noise, or Rayleigh fading, for example). However, when equipment failures and malfunctions are involved *in addition to the above factors*, it must be recognized that one has to deal with a tremendous variety of electronic components—tubes, transistors, resistors, capacitors, relays, teletype machines, paper tape punches, readers and transports—each with its own peculiar failure modes. Moreover, it becomes necessary to distinguish intermittent malfunctions of these components and devices from catastrophic, permanent failures. Not only are the effects on system performance of intermittent and permanent failures quite different, but when considered over the whole range of component types and conditions of use, these two types of statistics appear to be quite uncorrelated.

A second set of considerations will be described next that although they are less apparent than those described above, are probably even more significant in relation to the problem at hand. It must be realized that equipment malfunctions (whether temporary or permanent) can have widely disparate effects on the structure of the the transmitted (or received message). It is customary in discussing *channel* errors to consider either a symmetric crossover situation (probability of a *zero* being received as a *one* = probability of a *one* being received as a *zero*), or the erasure situation (*zeros* or *ones* being received as blanks). Both situations are pertinent as well to the equipment malfunction problem—fortunately these are innately tractable mathematically. But the problem does not end there. Equipment malfunctions can also result in a complicated variety of synchronization errors which possess very widely different effects on message reliability. Basically, all of these types involve incorrect interpretations by the receiver, or the transmitter, *or both* the receiver and transmitter, as to the structure of the message.

It is of course necessary when receiving data in a block code, that the receiver know:

(1) Where a code word begins and ends, and

(2) Which digits are information digits and which are the check digits.

In systems employing variable coding (for the purpose of matching time-varying channel statistics), it is also necessary that the receiver and transmitter be kept continually in agreement as to the actual code in use at any one time.

The achievement of this kind of agreement in timing (synchronization of transmitter and receiver) is generally obtained by means of stable clocks and periodic synchronization signals between transmitter and receiver. However, when equipment malfunctions are taken into account then one must also consider the possibility that the receiver will misinterpret the information it needs for items (1) and/or (2) above, even though it is receiving completely valid data. For example, the timing counter in the receiver, which separates information digits and check digits in a code word, may commit an error and cause a corresponding misinterpretation of these digits. Or, the receiver's block length counter may malfunction with the result that part of a received message may be ignored, or even that digits belonging to the next block will be misinterpreted as belonging to the present block. The details of such effects will depend very strongly on the precise logical structure used in the receiver's decoding equipment.

As if this were not bad enough, similar things can go wrong in the encoding process at the transmitter. Again, we have to deal with timing counters that determine the structure of the transmitted message. Let us assume that the transmitter encoder accepts blocks of k digits at a time, and encodes these blocks into (longer) blocks of n digits, including redundancy (for error control and perhaps for synchronization). Encoder malfunctions can then cause the transmitter to ignore portions of an uncoded k-digit block during the encoding process. Or, check digits may be dropped from the transmitted message, resulting in the transmission of a block of fewer than the required n digits. The action taken by the receiver on receipt of such a short block will depend on what consideration has been given to this eventuality in the design of the decoder. At best, one can only hope that a repeat transmission (in a decision feedback communication system) will be requested by the receiver. In a one-way link, the received block is virtually certain to be entirely unusable.

3. CONCLUSIONS

Reflection regarding the factors discussed above, and also experience attained with error-correction systems, seems to indicate quite strongly that the most important causes of errors in such systems lies in loss of synchronization at some point in the system. This is particularly true when the basic block length of the message is long, since a single "dropped" or "added" digit can then result in the incorrect decoding of a large number of message digits.

Synchronization questions have been largely ignored in most discussions of coding until very recently. In relation to the problem of over-all system reliability, they cannot validly be ignored, since it is completely unrealistic to assume that the timing and counting equipment at the terminals will be inherently much more reliable than the information-processing components themselves.

A close enough look has been taken at this problem to suggest that no generally applicable technique exists that will permit one to decide how over-all transmission reliability is going to be affected by the addition of error-correction terminal equipment. Such conclusions can only be arrived at by very detailed study of the failure modes and statistics of all the components employed in this equipment, with detailed consideration of how each type of failure will affect the transmitted information, the received information, and especially the message structure. Conclusions reached for one particular set of equipment are likely to be completely invalid for equipment that differs only slightly from the original. This is so mainly because of the overwhelming importance of providing protection against synchronization errors. Relatively minor changes in the design of timing counters and shift registers, for example, can result in large improvements in their reliability.

For the same reason, it is possible to recommend that increased consideration be given to the study of techniques of coding, modulation and equipment design for the purpose of providing extremely accurate and reliable transmitter-receiver synchronization.

REFERENCES

1. Elspas, Bernard, "Design and Instrumentation of Error-Correcting Codes," Interim Technical Report, Contract AF 30(602)-2327, RADC TR 61-259, SRI Project 3318, Stanford Research Institute, Menlo Park, California (October 1961).

2. Elspas, Bernard, and Robert A. Short, "A Table of Indices for Polynomials over *GF*(2)," Supplement No. 1 to Interim Technical Report, Contract AF 30(602)-2327, RADC TR 61-259, SRI Project 3318, Stanford Research Institute, Menlo Park, California (October 1961).

3. Shannon, C. E., and W. Weaver, *Mathematical Theory of Communication*, (University of Illinois Press, Urbana, Illinois, 1949).

4. Elias, P., "Coding for Noisy Channels," *IRE Convention Record*, Part 4, pp. 37-46 (1955).

5. Elias, P., "Coding for Two Noisy Channels," pp. 61-74, *Information Theory*, Colin Cherry, Ed. (Academic Press, New York, N.Y., 1956).

6. Shannon, P. E., Certain Results in Coding Theory for Noisy Channels," *Inf. and Control* **1**, pp. 6-25 (1957).

7. Wozencraft, J. M., and B. Reiffen, *Sequential Decoding* (The Technology Press of M.I.T. and John Wiley and Sons, Inc., New York, N.Y., 1961).

8. Bose, R. C., and D. C. Ray-Chaudhuri, "On a Class of Error Correcting Binary Group Codes," *Inf. and Control* **3**, pp. 68-79 (1960).

9. Bose, R. C., and D. C. Ray-Chaudhuri, "Further Results on Error Correcting Binary Group Codes," *Inf. and Control* **3**, pp. 279-290 (1960).

10. Abramson, N. M., "A Class of Systematic Codes for Non-Independent Errors," *IRE Trans. PGIT*-**5**, pp. 150-157 (1959).

11. Fire, P., "A Class of Multiple-Error-Correcting Binary Codes for Non-Independent Errors," Sylvania Report RSL-E-2, Sylvania Reconnaissance Systems Laboratory, Mountain View, California (1959).

12. Melas, C. M., "A New Group of Codes for Correction of Dependent-Errors in Data Transmission," *IBM J. Research Develop.* **4**, pp. 58-65 (1960).

13. Elspas, Bernard, "A Note on *P*-nary Adjacent-Error-Correcting Codes," *IRE Trans. PGIT*-**6**, pp. 13-15 (1960).

14. Corr, F., "Multiple Burst Detection," *Proc. IRE* **49**, p. 1337 (August 1961).

15. Upensky, J. V., and M. A. Heaslet, *Elementary Number Theory*, (McGraw-Hill Book Company, New York, N.Y., 1939).

16. Jacobson, N., *Lectures in Abstract Algebra, Vol.* I (D. VanNostrand Co., New York, 1951).

17. Reed, I. S., and G. Solomon, "Polynomial Codes over Certain Finite Fields," *J. Soc. Indust. Appl. Math.* **8**, pp. 300-304 (1960).

18. Elias, P., "Error-Free Coding," *Trans. IRE PGIT*-**4**, pp. 29-37 (1954).

19. Slepian, D., "Further Theory of Group Codes," *Bell System Tech. J.* **39**, p. 1219 (1960).

20. Peterson, W. W., *Error-Correcting Codes*, (John Wiley and Sons, Inc., New York, N.Y., 1961) p. 81.

21. Kautz, W. H., "A Class of Multiple-Error-Correcting Codes for Data Transmission and Recording," Tech. Report 5, Contract DA 36-039-SC-66381, SRI Project 2124, Stanford Research Institute, Menlo Park, California (1959).

22. Calingaert, P., "Two-Dimensional Parity Checking," *J. Assoc. Computing Machinery* **8**, pp. 186-200 (1961).

23. Rubinoff, M. "*N*-Dimensional Codes for Detecting and Correcting Multiple Errors," *Communs. ACM* 4, pp. 545-551 (1961).

24. Bennion, D. R., "MAD-Resistance Type Magnetic Shift Register," paper presented at Conference on Non-Linear Magnetics and Magnetic Amplifiers, Philadelphia, Pennsylvania, 26-28 October 1960.

25. Kamasi, T., "Optimum Shortened Cyclic Codes for Burst-Error Correction," *IRE Trans. PGIT* (in press).

26. Klein, R. D., "Reliability of Coded and Uncoded Binary Messages as a Function of the Rate of Symbol Transmission," paper presented at WESCON, San Francisco, California, August 1961.

STANFORD
RESEARCH
INSTITUTE

MENLO PARK
CALIFORNIA

Regional Offices and Laboratories

Southern California Laboratories
820 Mission Street
South Pasadena, California

Washington Office
808 17th Street, N.W.
Washington 5, D.C.

New York Office
270 Park Avenue, Room 1770
New York 17, New York

Detroit Office
The Stevens Building
1025 East Maple Road
Birmingham, Michigan

European Office
Pelikanstrasse 37
Zurich 1, Switzerland

Japan Office
911 Iino Building
22, 2-chome, Uchisaiwai-cho, Chiyoda-ku
Tokyo, Japan

Representatives

Honolulu, Hawaii
Finance Factors Building
195 South King Street
Honolulu, Hawaii

London, England
19 Upper Brook Street
London, W. 1, England

Milan, Italy
Via Macedonio Melloni 40
Milano, Italy

London, Ontario, Canada
P.O. Box 782
London, Ontario, Canada